Fig. 1

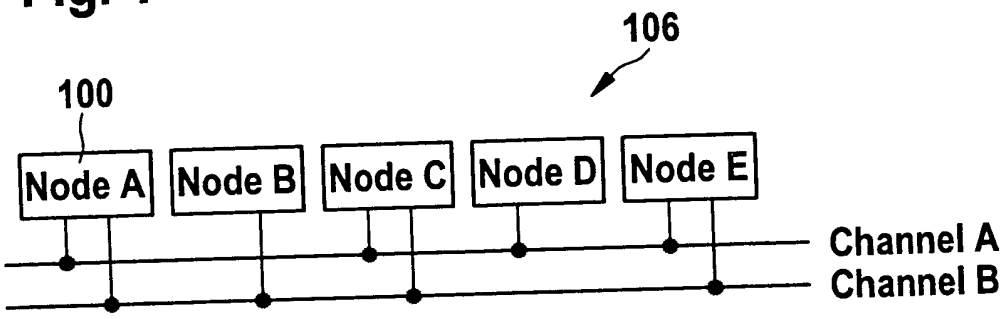


Fig. 2

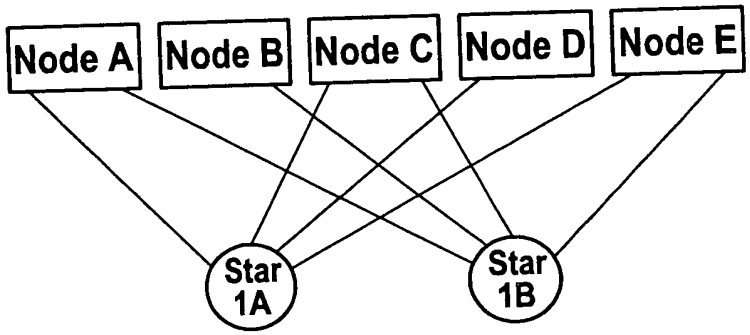


Fig. 3

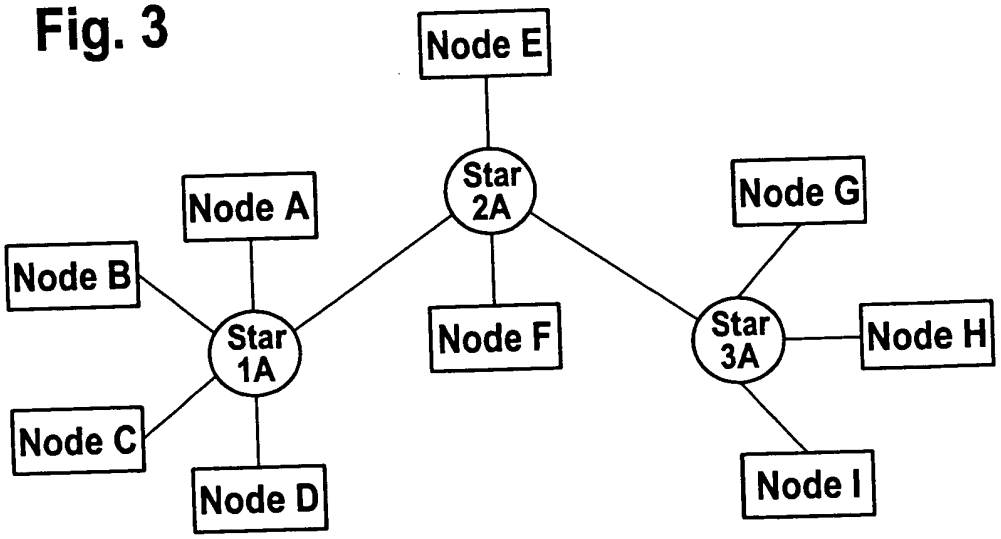


Fig. 4

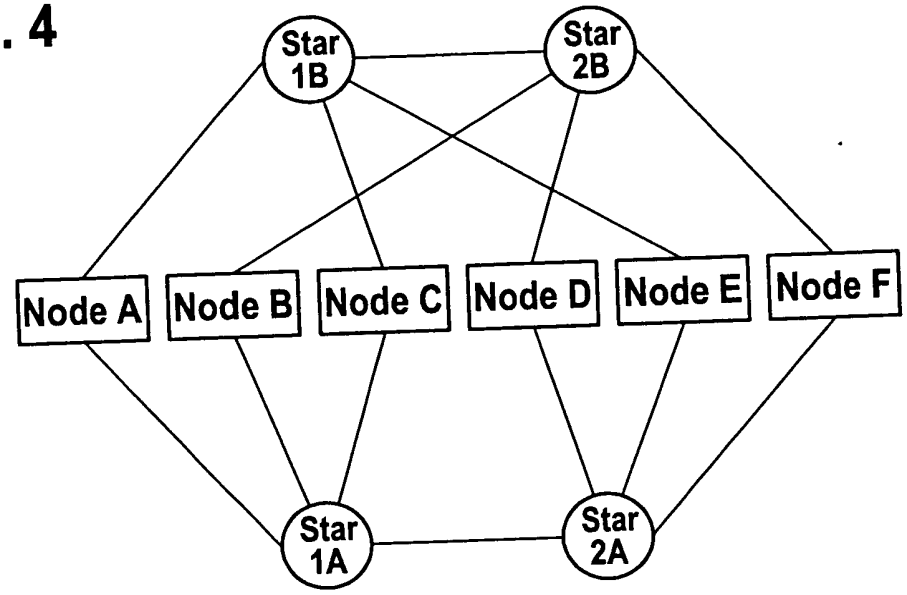

Star 1B
Star 2B
Node A
Node B
Node C
Node D
Node E
Node F
Star 1A
Star 2A

Fig. 5

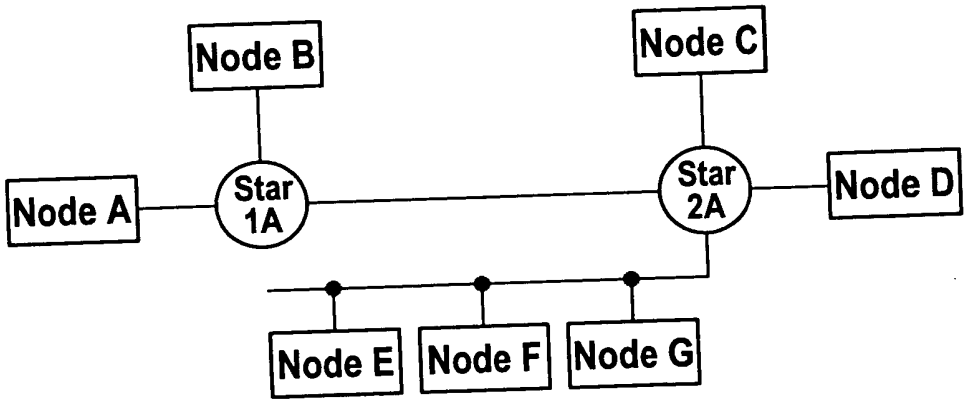

Node B
Node C
Node A
Star 1A
Star 2A
Node D
Node E
Node F
Node G

Fig. 6

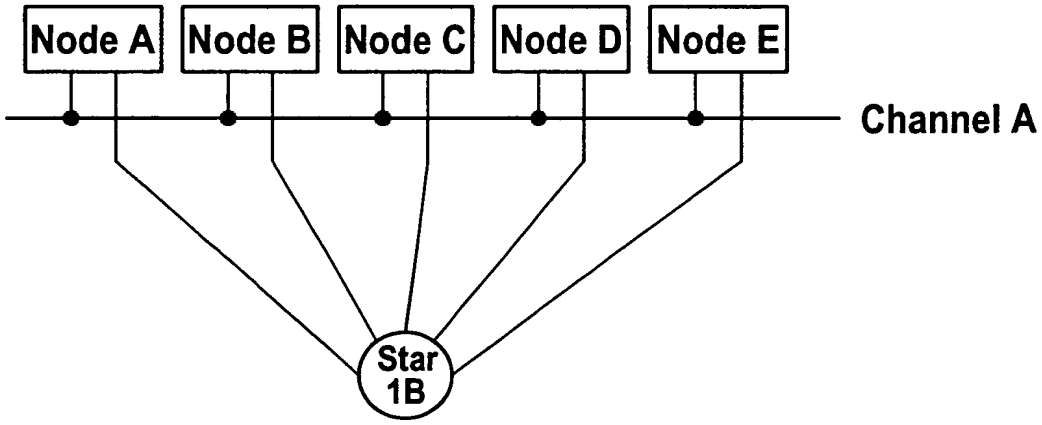

Node A
Node B
Node C
Node D
Node E
Channel A
Star
1B

Fig. 7

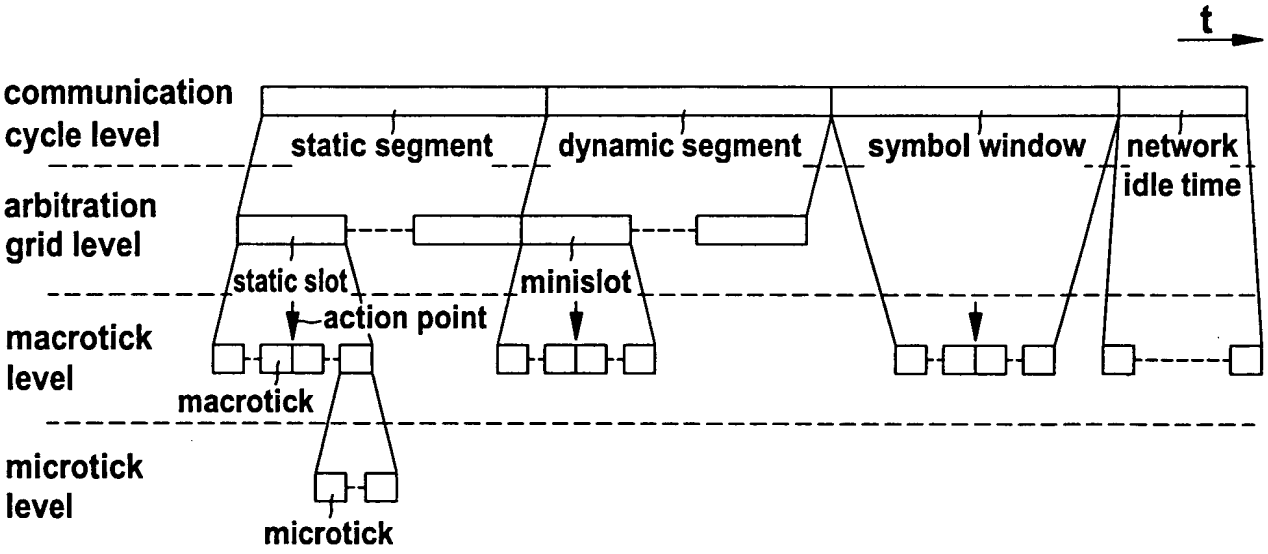

t
communication
cycle level
static segment
dynamic segment
symbol window
network
idle time
arbitration
grid level
static slot
minislot
action point
macrotick
level
macrotick
microtick
level
microtick

Fig. 8

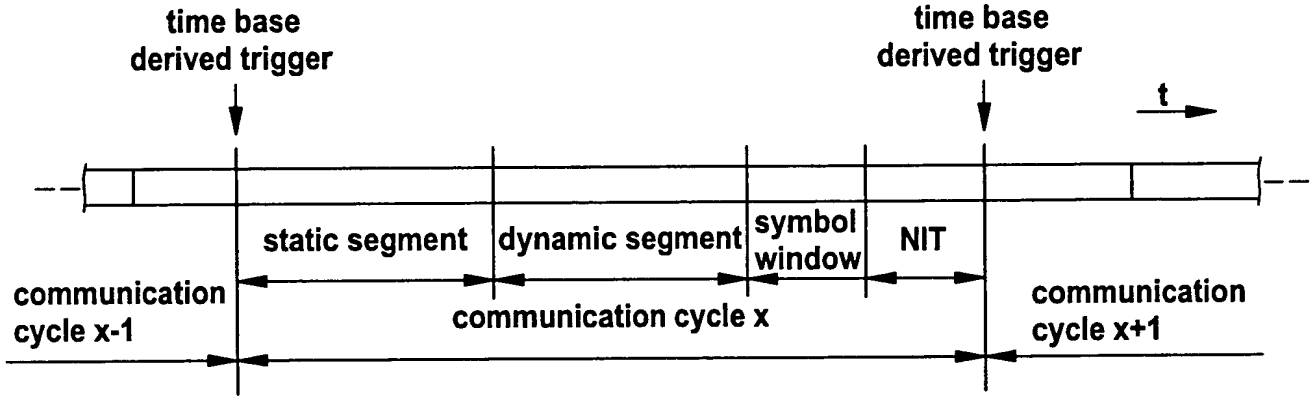


Fig. 9

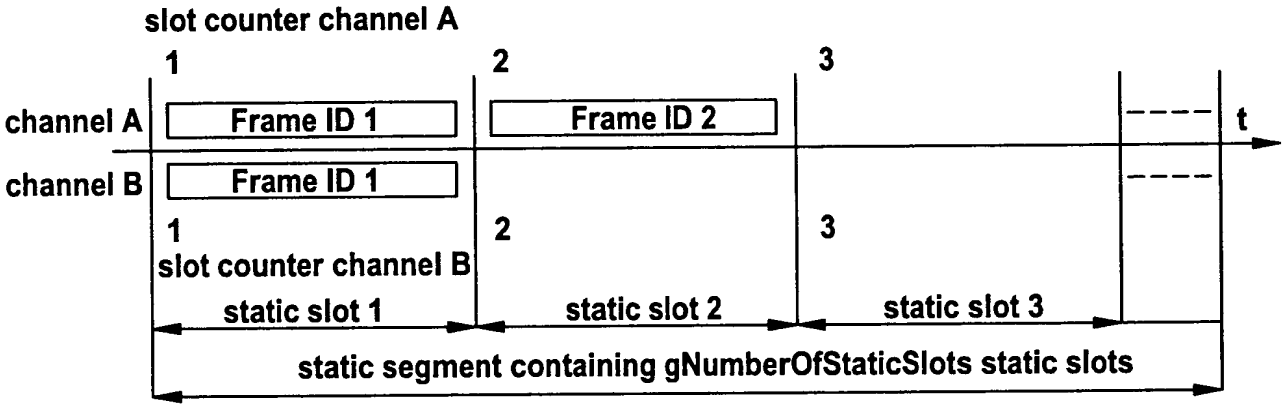


Fig. 10

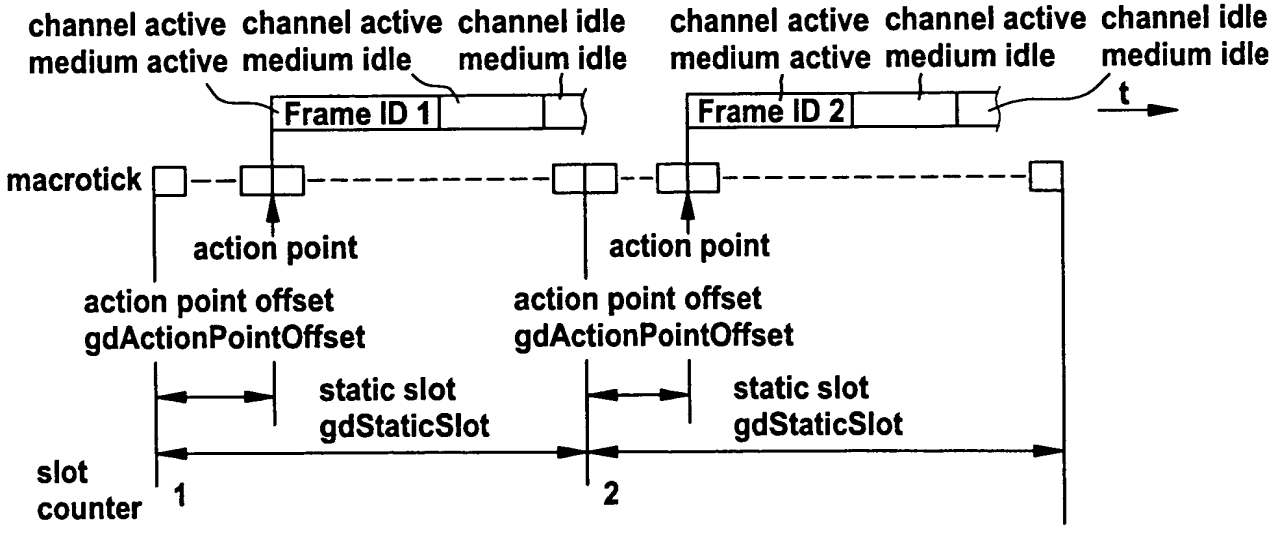


Fig. 11

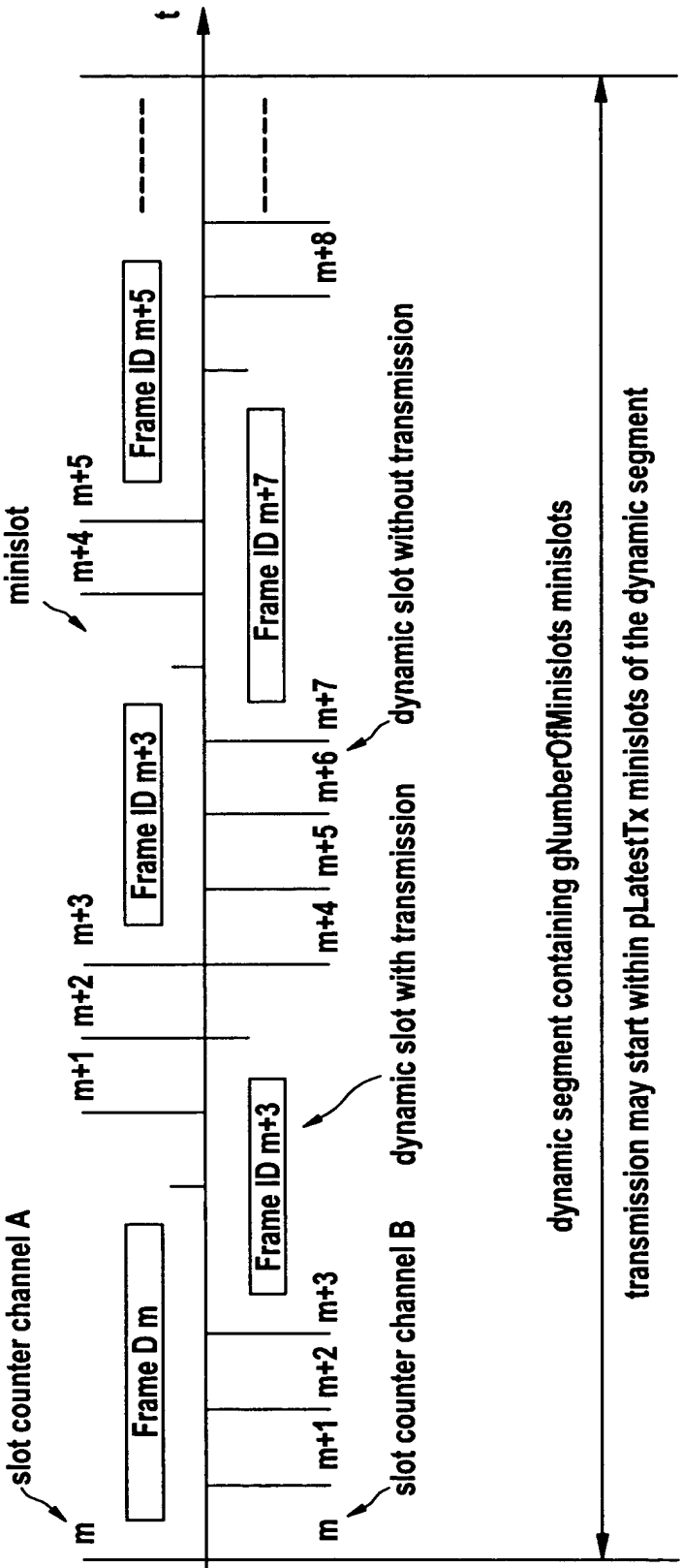

t
channel A
channel B
slot counter channel A
slot counter channel B
m
m+1
m+2
m+3
m+4
m+5
m+6
m+7
m+8
minislot
Frame D m
Frame ID m+3
Frame ID m+5
Frame ID m+7
dynamic slot with transmission
dynamic slot without transmission
dynamic segment containing gNumberOfMinislots minislots
transmission may start within pLatestTx minislots of the dynamic segment

Fig. 12

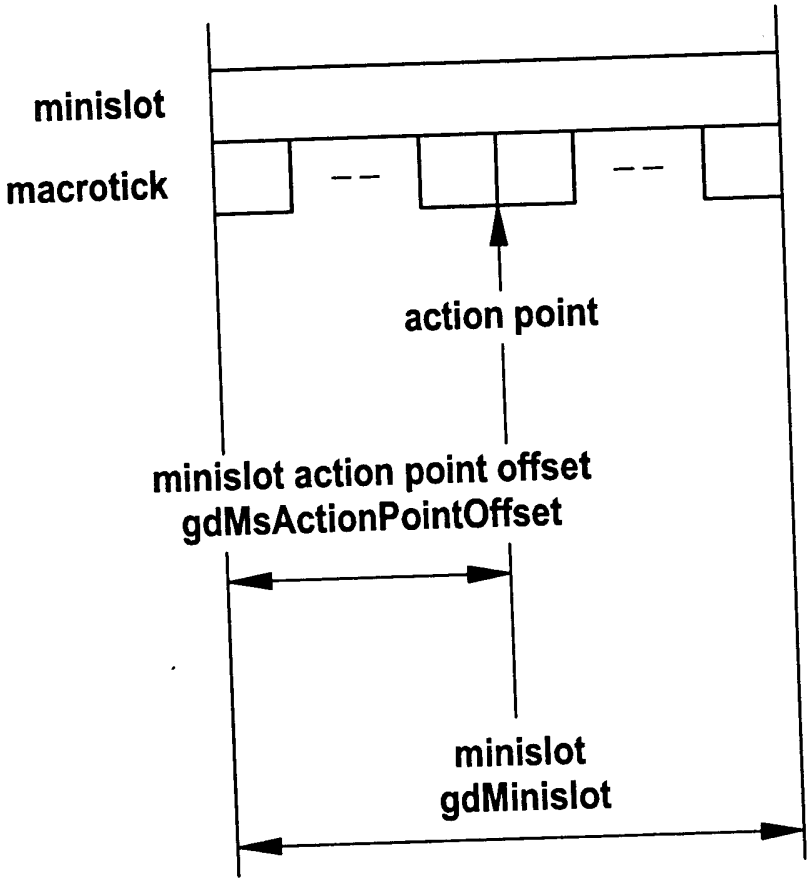

minislot
macrotick
action point
minislot action point offset
gdMsActionPointOffset
minislot
gdMinislot

Fig. 13

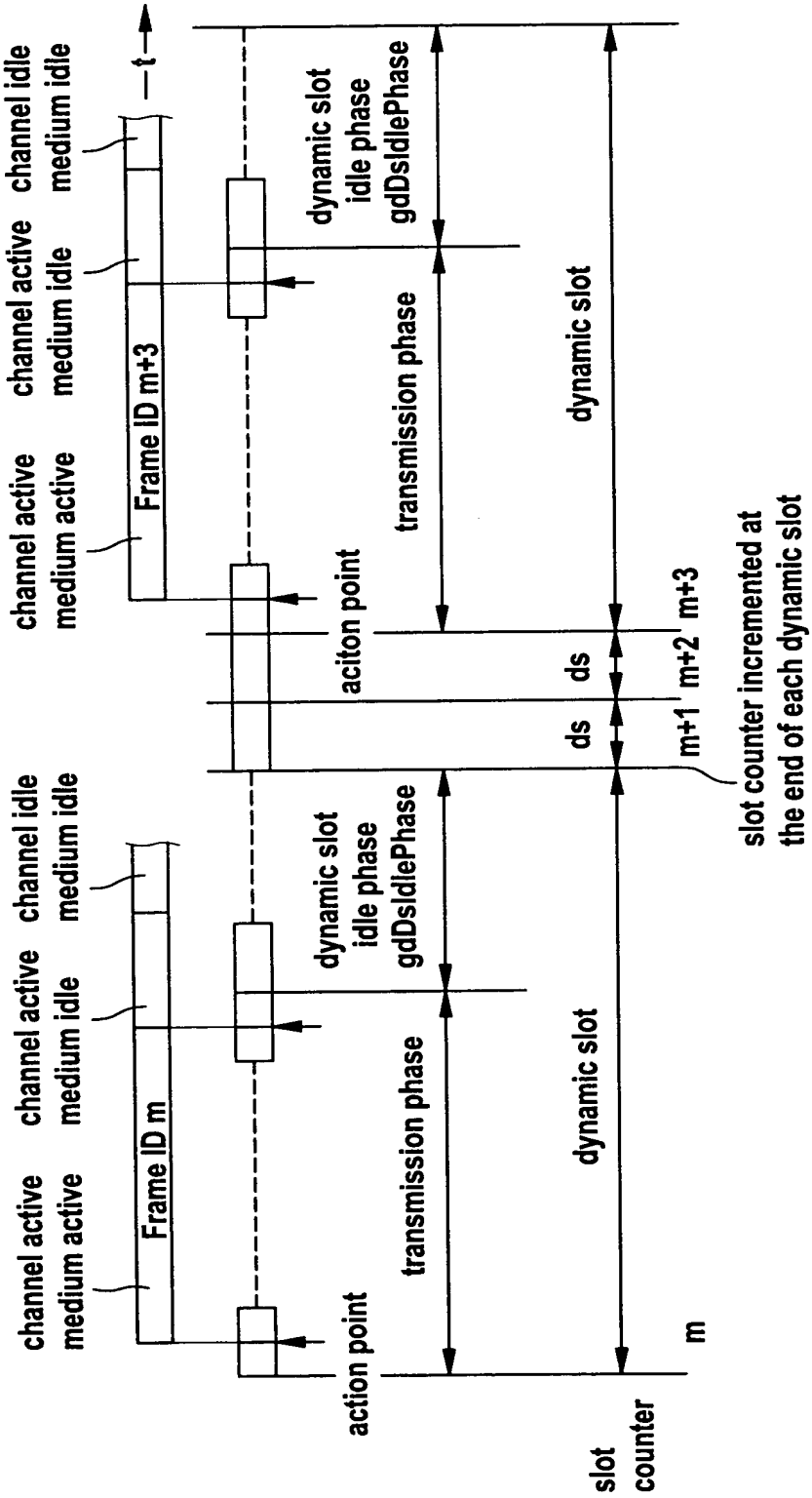
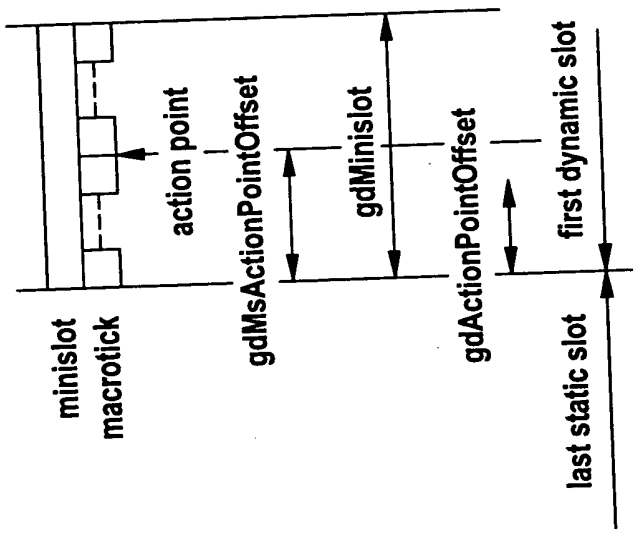

channel active
medium active
channel active
medium idle
channel idle
medium idle
Frame ID m
Frame ID m+3
dynamic slot
idle phase
gdDsIdlePhase
transmission phase
dynamic slot
action point
aciton point
ds
ds
m
m+1
m+2
m+3
slot counter
slot counter incremented at
the end of each dynamic slot
t

Fig. 14


minislot
macrotick
action point
gdMsActionPointOffset
gdMinislot
gdActionPointOffset
first dynamic slot
last static slot

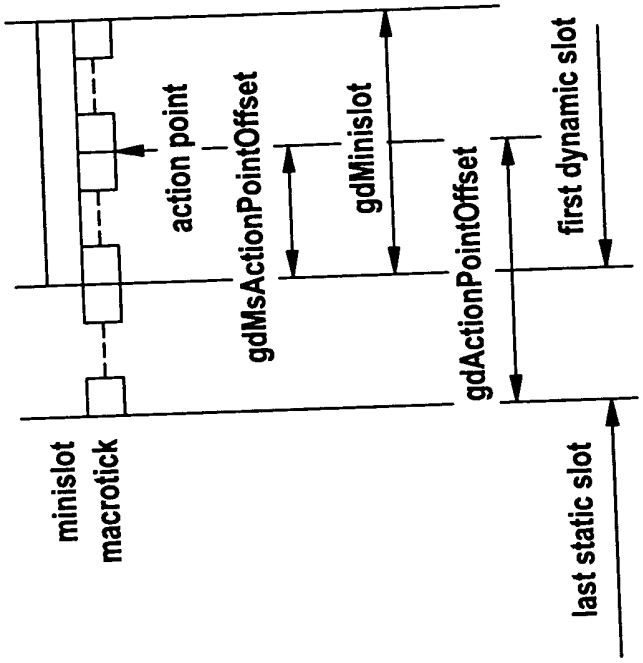

minislot
macrotick
action point
gdMsActionPointOffset
gdMinislot
gdActionPointOffset
first dynamic slot
last static slot

Fig. 15

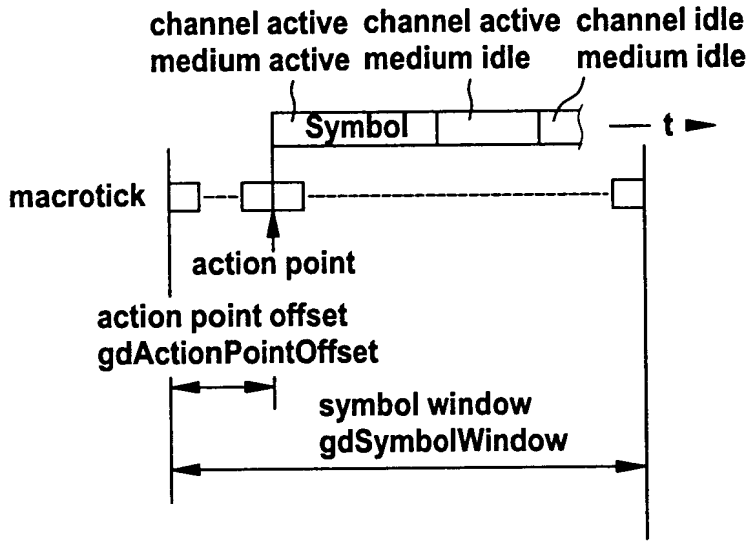

channel active
medium active
channel active
medium idle
channel idle
medium idle
Symbol
t
macrotick
action point
action point offset
gdActionPointOffset
symbol window
gdSymbolWindow

Fig. 16

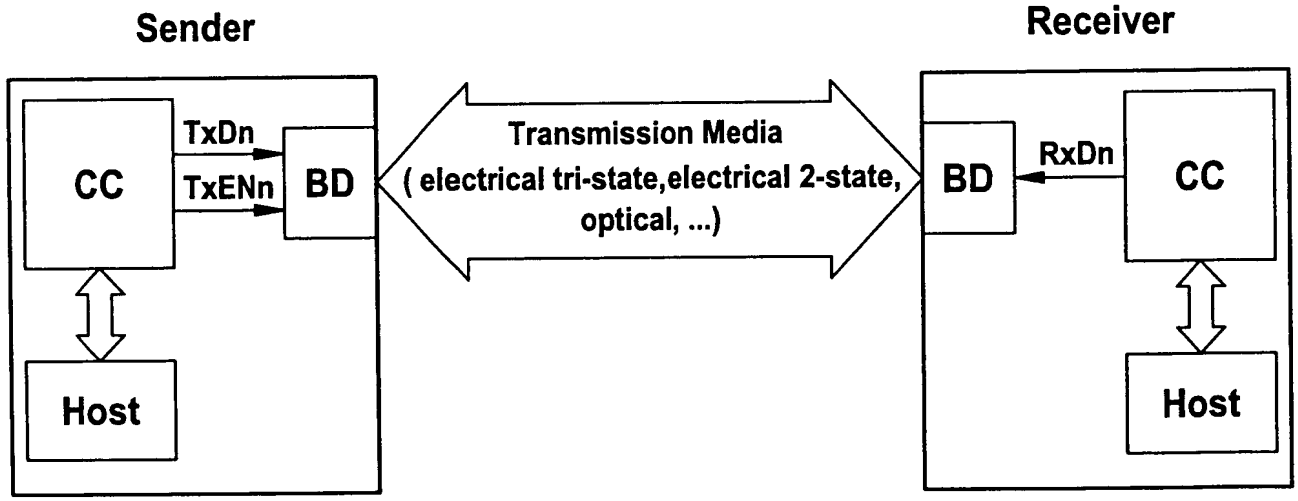

Sender
Receiver
CC
TxDn
TxENn
BD
Transmission Media
(electrical tri-state,electrical 2-state,
optical, ...)
BD
RxDn
CC
Host
Host

Fig. 17

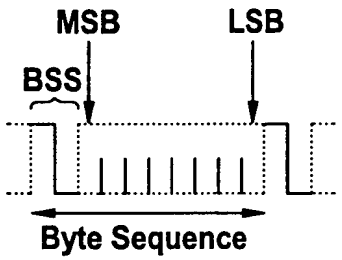


Fig. 18

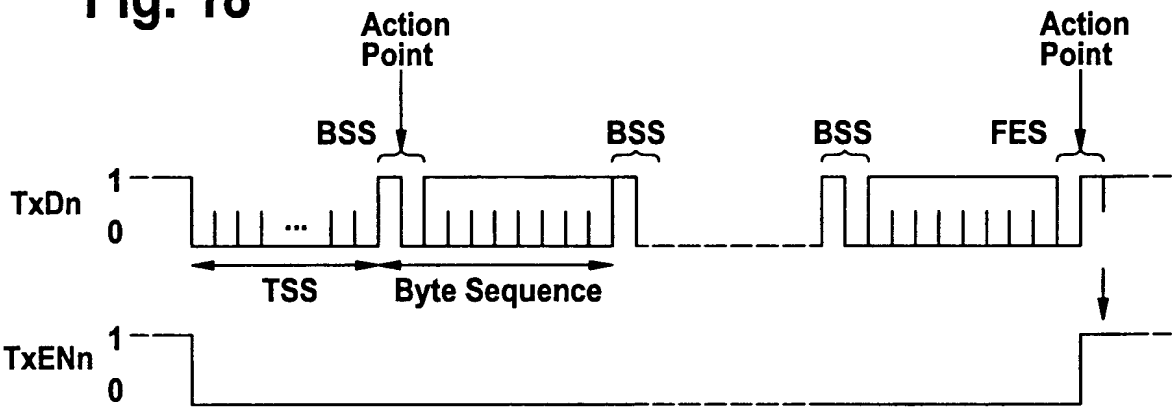


Fig. 19

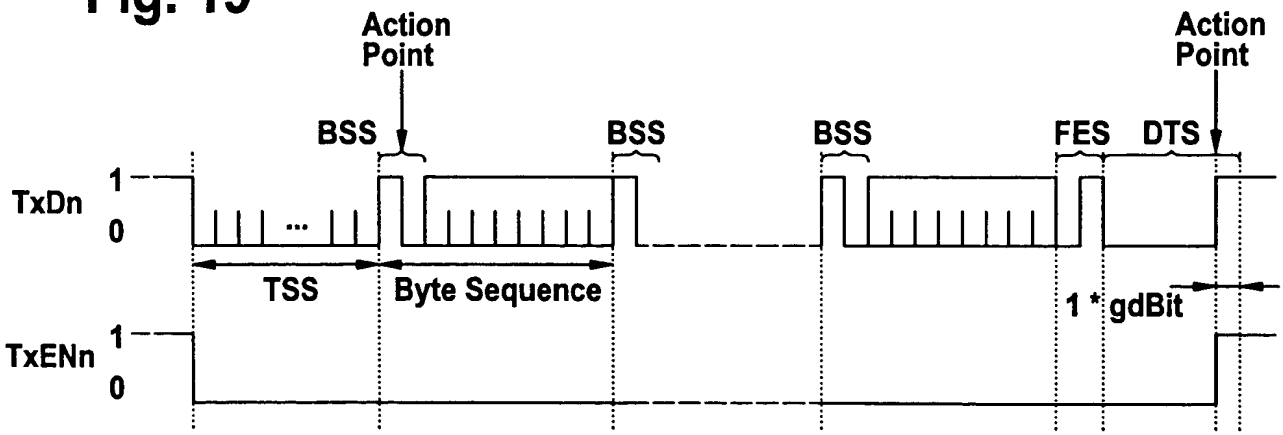


Fig. 20

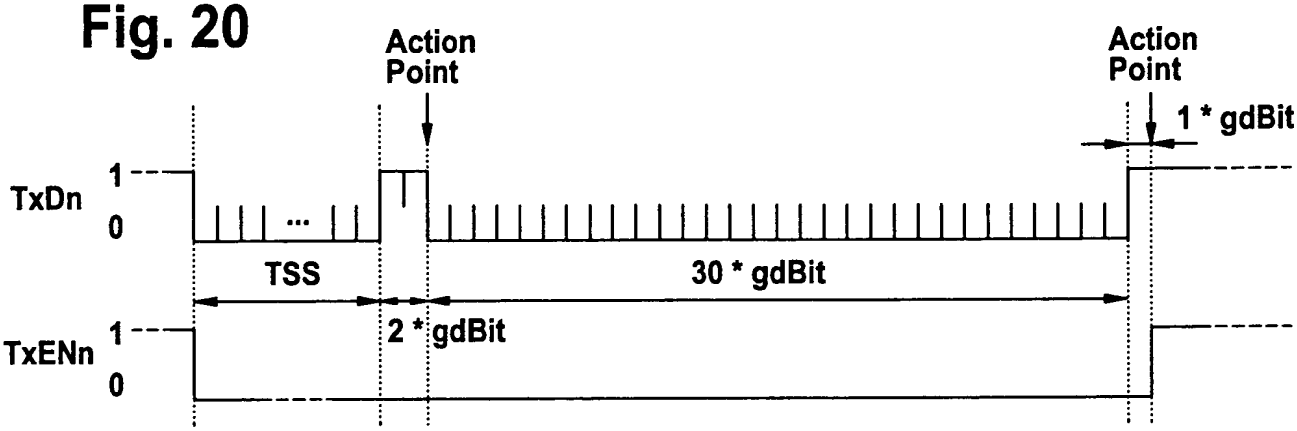


Fig. 21

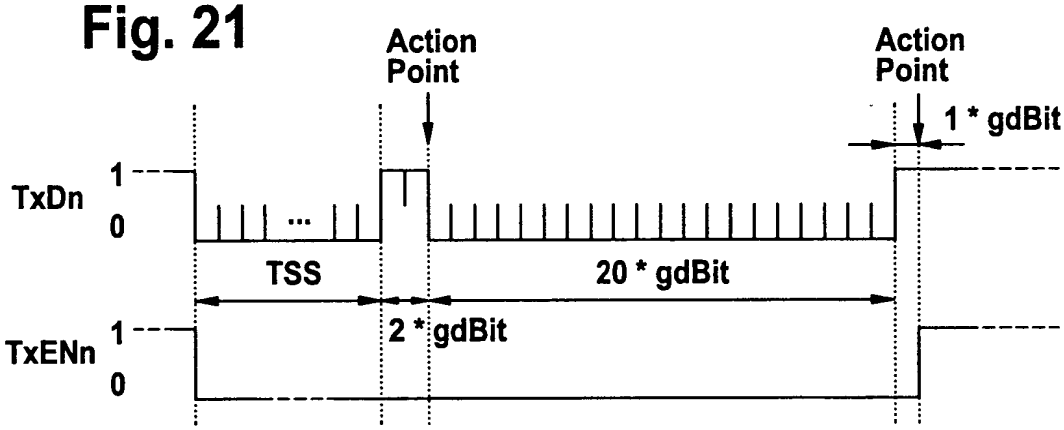


Fig. 22

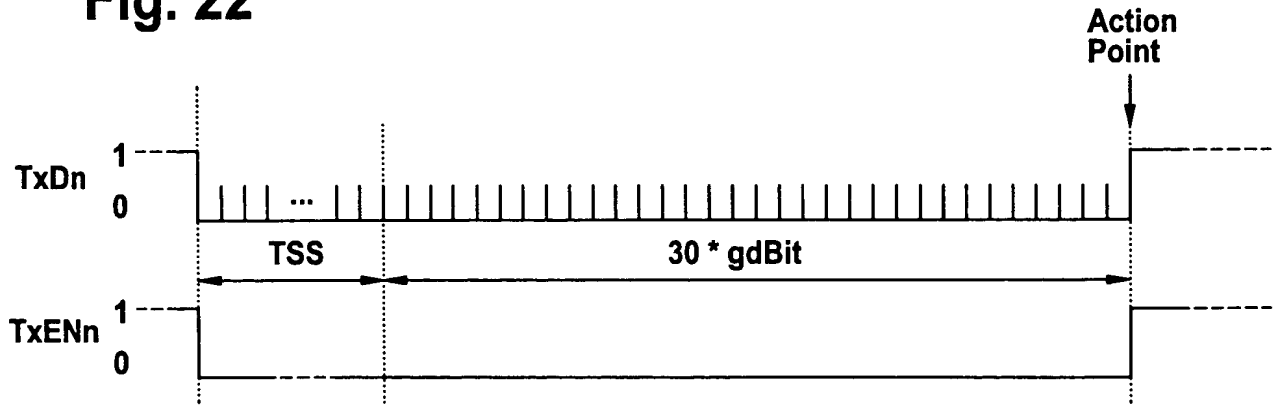


Fig. 23

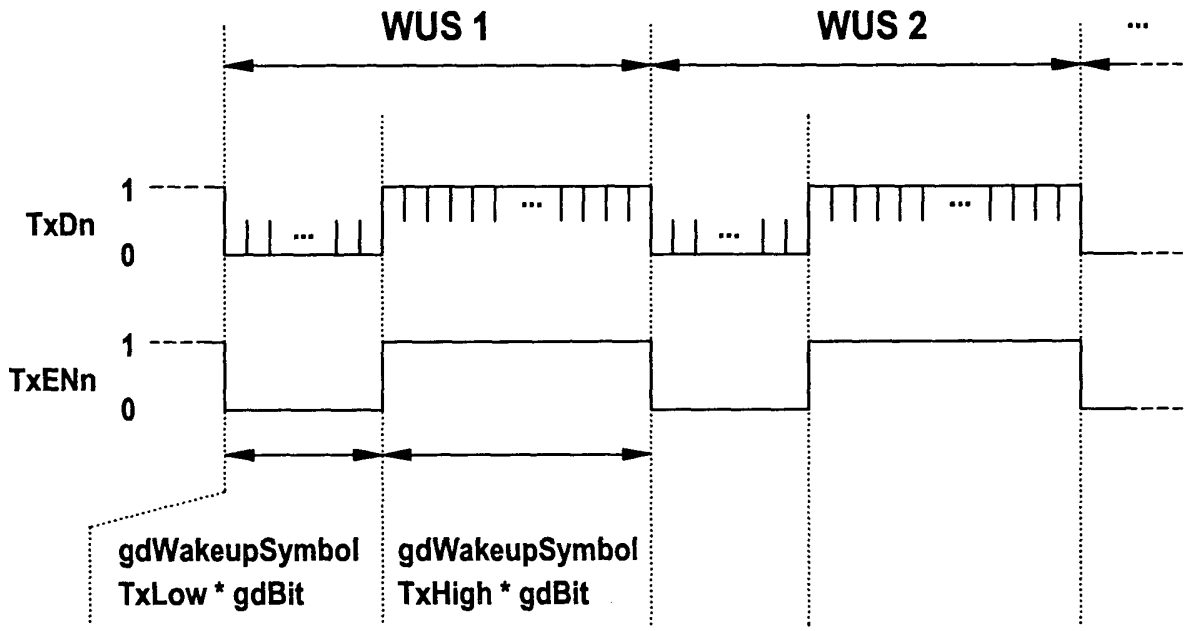

WUS 1
WUS 2
...
TxDn
1
0
TxENn
1
0
gdWakeupSymbol
TxLow * gdBit
gdWakeupSymbol
TxHigh * gdBit

Fig. 24

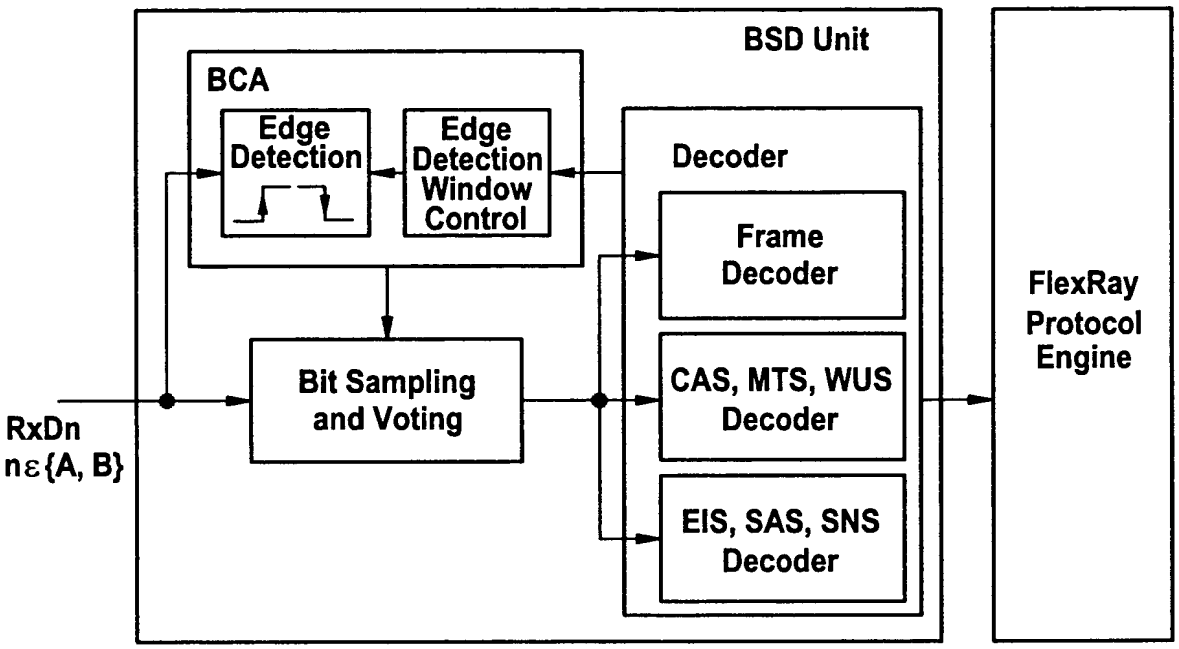

BSD Unit
BCA
Edge Detection
Edge Detection Window Control
Decoder
Frame Decoder
CAS, MTS, WUS Decoder
EIS, SAS, SNS Decoder
Bit Sampling and Voting
RxDn
nε{A, B}
FlexRay Protocol Engine

Fig. 25

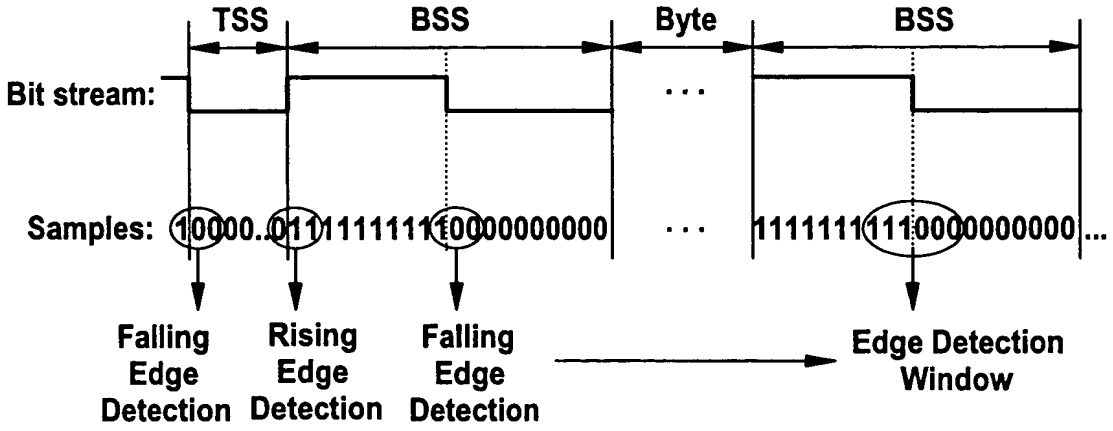

TSS
BSS
Byte
BSS
Bit stream:
Samples:
10000..011111111110000000000
1111111111000000000...
Falling Edge Detection
Rising Edge Detection
Falling Edge Detection
Edge Detection Window

Fig. 26

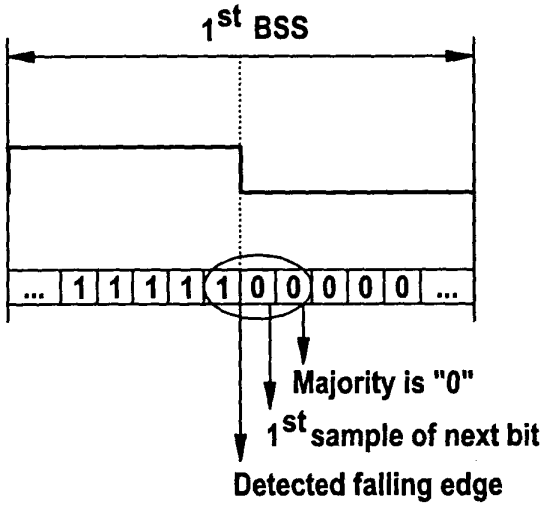

1st BSS
... 1 1 1 1 1 0 0 0 0 0 ...
Majority is "0"
1st sample of next bit
Detected falling edge

Fig. 27

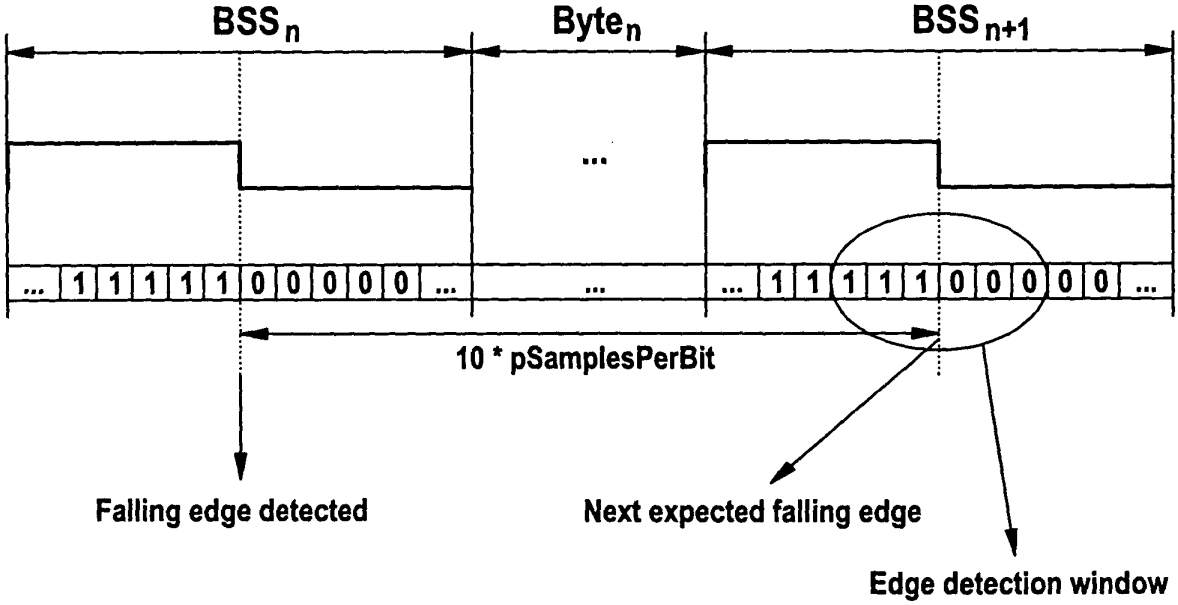

BSSn
Byten
BSSn+1
...
... 1 1 1 1 1 0 0 0 0 0 ...
... 1 1 1 1 1 0 0 0 0 0 ...
10 * pSamplesPerBit
Falling edge detected
Next expected falling edge
Edge detection window

Fig. 28

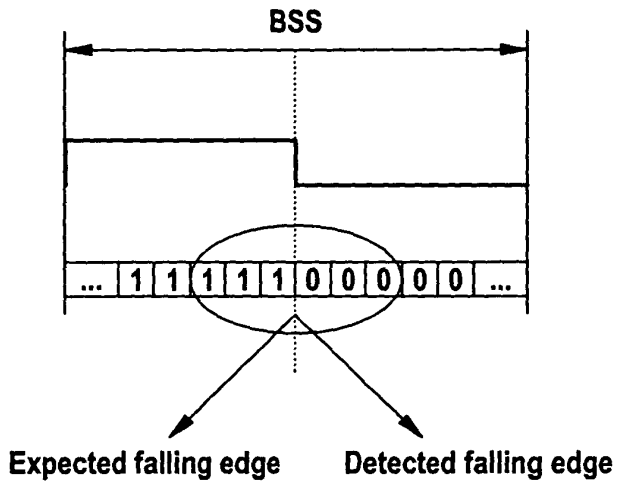


Fig. 29

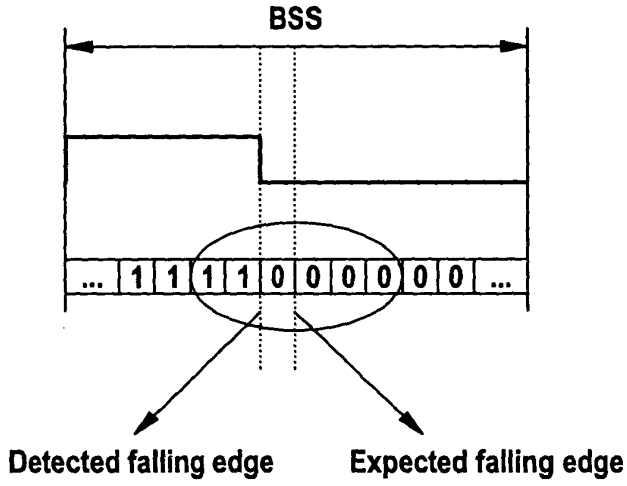


Fig. 30

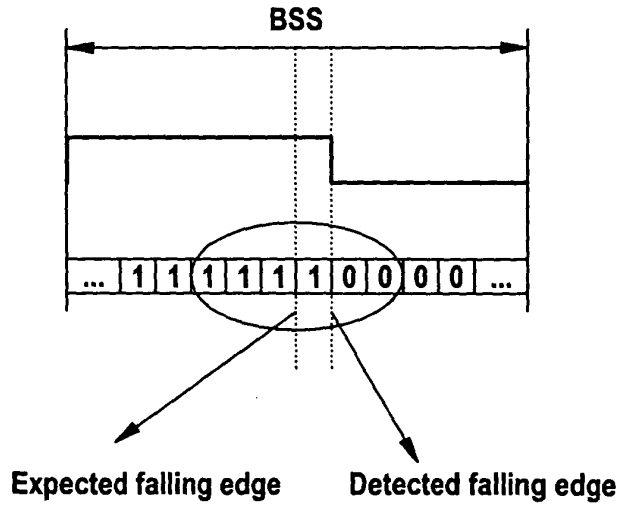


Fig. 31

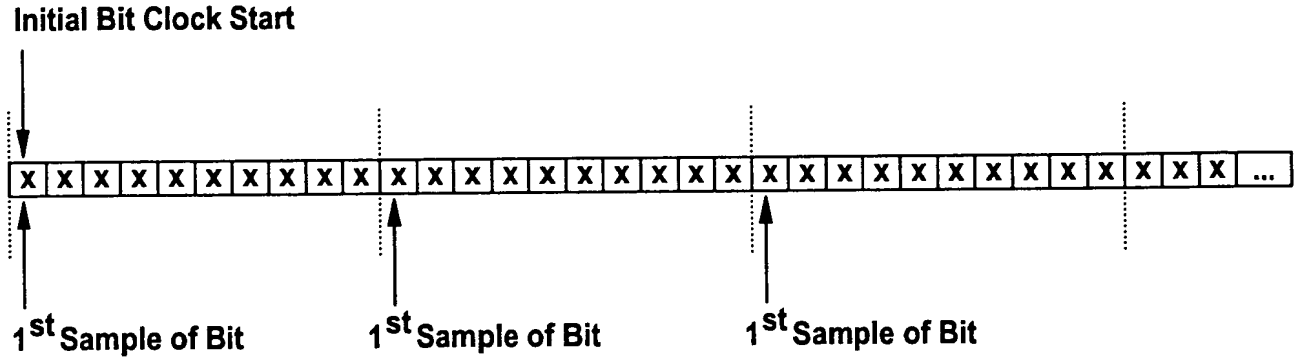


Fig. 32

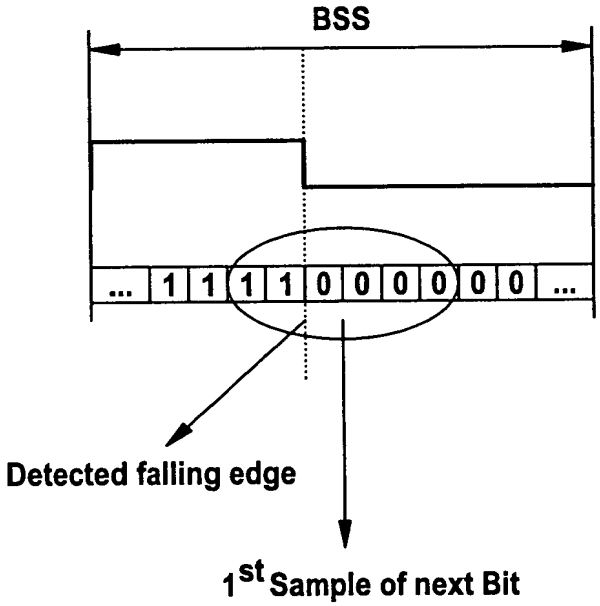


Fig. 33

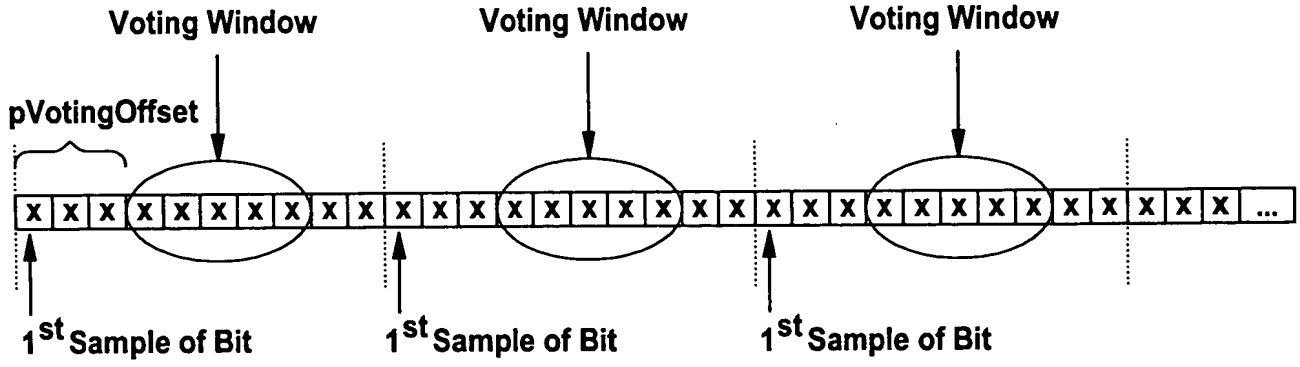


Fig. 34

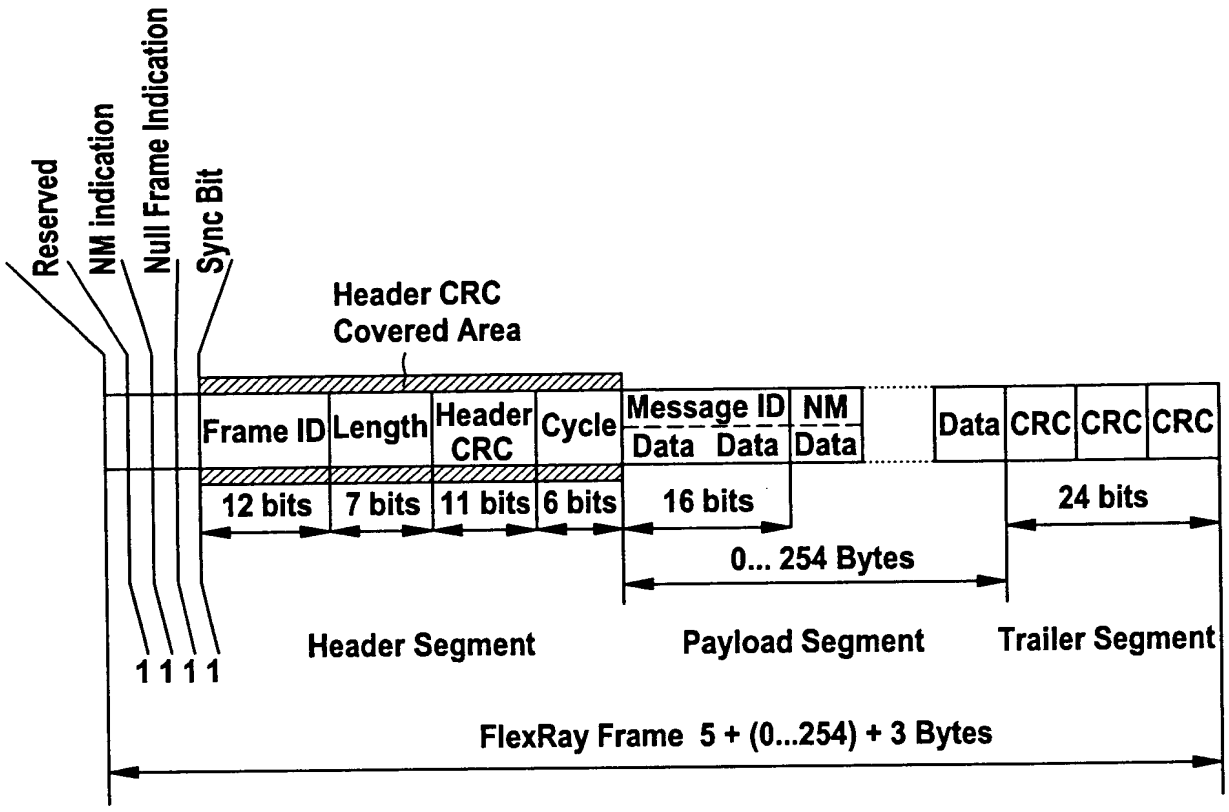

Reserved
NM indication
Null Frame Indication
Sync Bit
Header CRC
Covered Area
Frame ID
Length
Header CRC
Cycle
Message ID
Data
NM
Data
Data
CRC
CRC
CRC
12 bits
7 bits
11 bits
6 bits
16 bits
24 bits
0... 254 Bytes
1 1 1 1
Header Segment
Payload Segment
Trailer Segment
FlexRay Frame 5 + (0...254) + 3 Bytes

Fig. 35

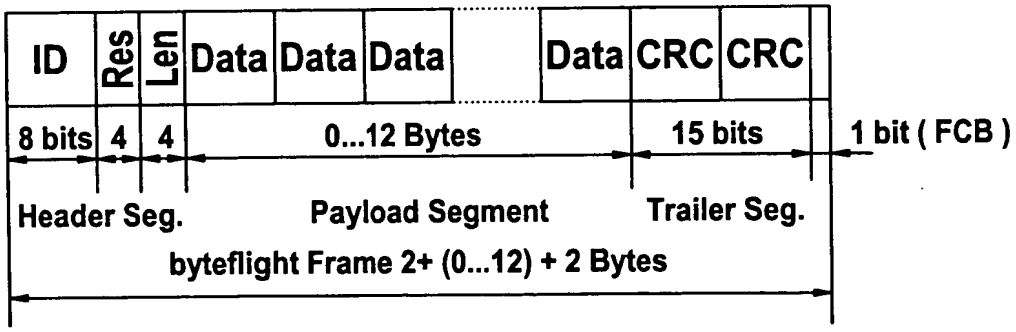

ID
Res
Len
Data
Data
Data
Data
CRC
CRC
8 bits
4
4
0...12 Bytes
15 bits
1 bit (FCB)
Header Seg.
Payload Segment
Trailer Seg.
byteflight Frame 2+ (0...12) + 2 Bytes

Fig. 36

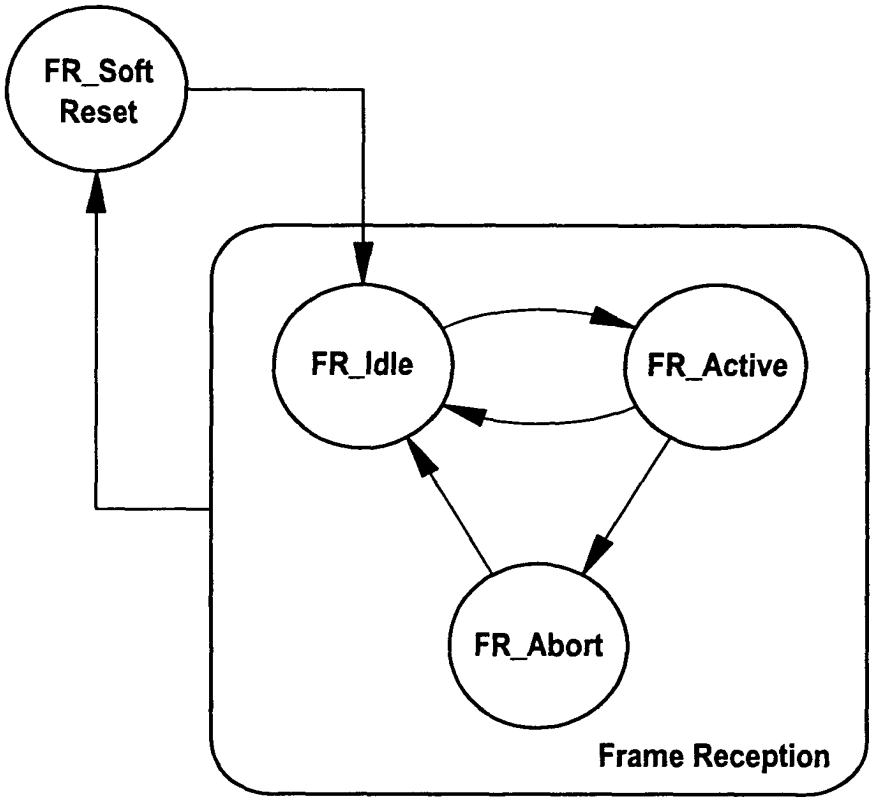

FR_Soft
Reset
FR_Idle
FR_Active
FR_Abort
Frame Reception

Fig. 37

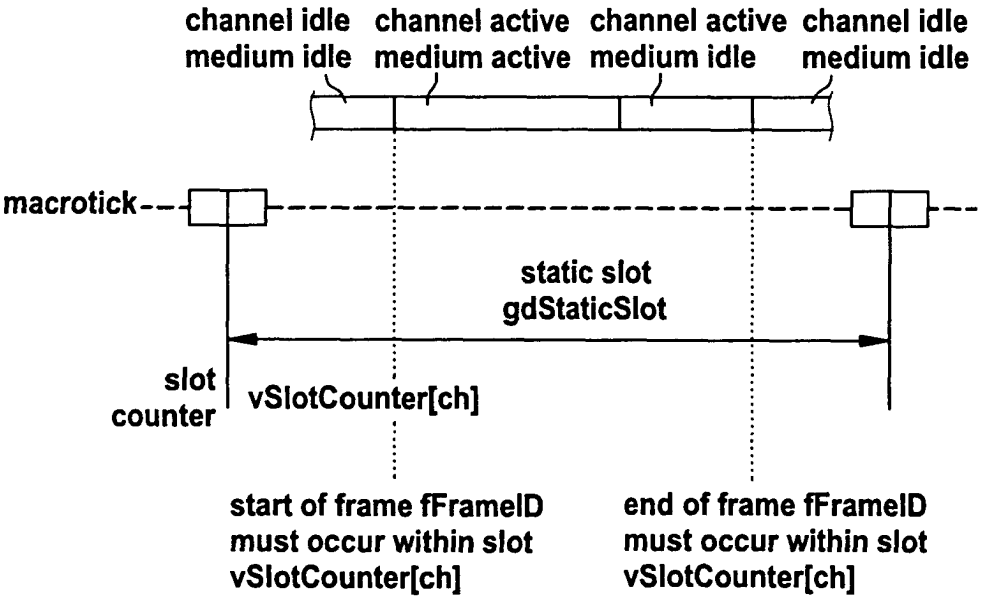

channel idle
medium idle
channel active
medium active
channel active
medium idle
channel idle
medium idle
macrotick
static slot
gdStaticSlot
slot
counter
vSlotCounter[ch]
start of frame fFrameID
must occur within slot
vSlotCounter[ch]
end of frame fFrameID
must occur within slot
vSlotCounter[ch]

Fig. 38

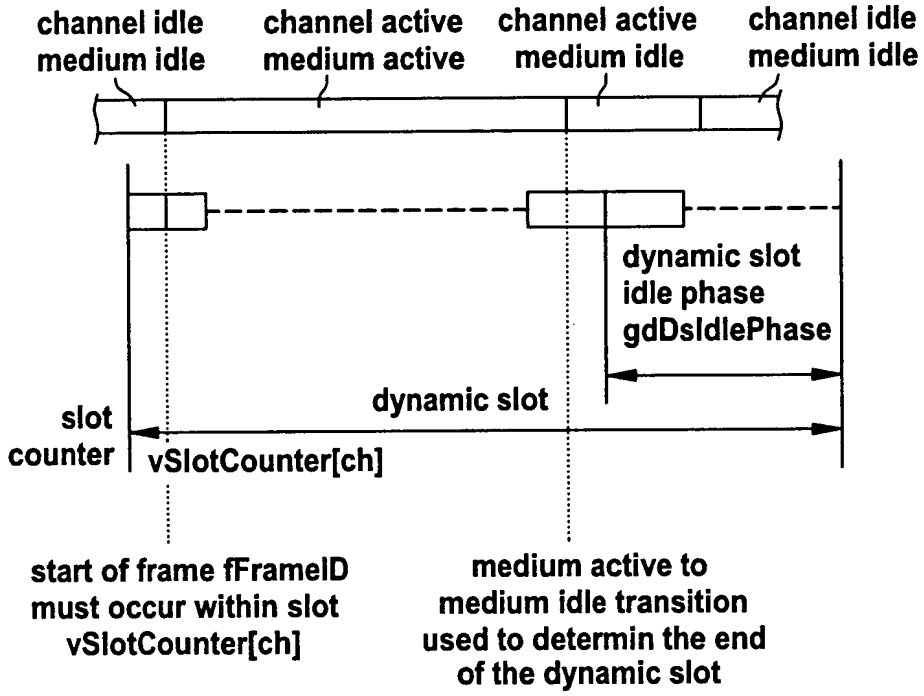

channel idle
medium idle
channel active
medium active
channel active
medium idle
channel idle
medium idle
dynamic slot
idle phase
gdDsIdlePhase
dynamic slot
slot
counter
vSlotCounter[ch]
start of frame fFrameID
must occur within slot
vSlotCounter[ch]
medium active to
medium idle transition
used to determin the end
of the dynamic slot

Fig. 39

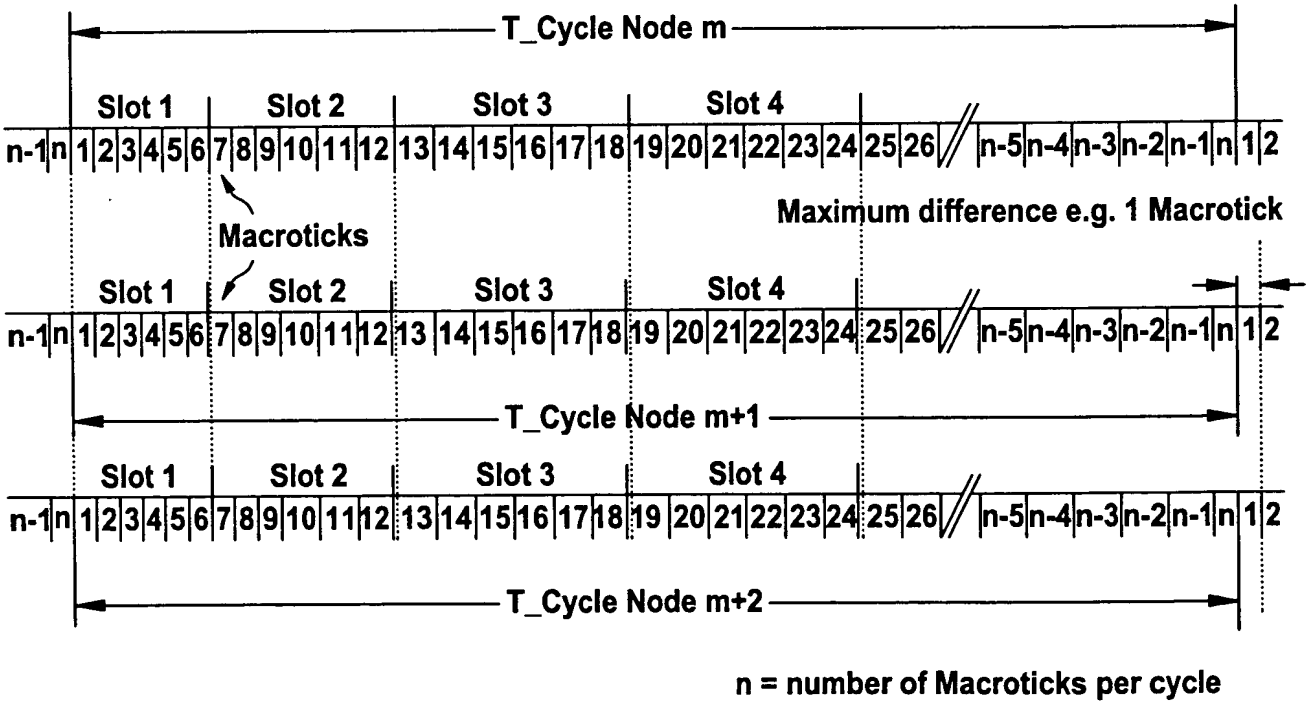

T_Cycle Node m
Slot 1
Slot 2
Slot 3
Slot 4
n-1 n 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 n-5 n-4 n-3 n-2 n-1 n 1 2
Macroticks
Maximum difference e.g. 1 Macrotick
T_Cycle Node m+1
T_Cycle Node m+2
n = number of Macroticks per cycle

Fig. 40

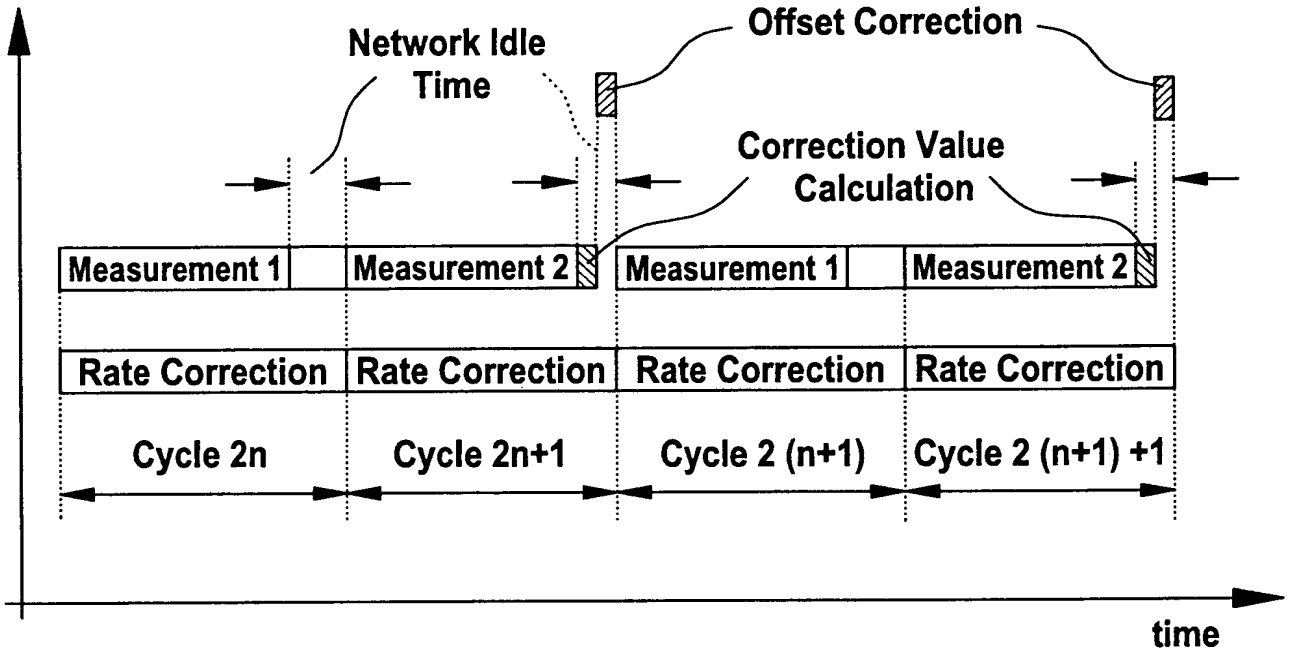

Network Idle Time
Offset Correction
Correction Value Calculation
Measurement 1
Measurement 2
Measurement 1
Measurement 2
Rate Correction
Rate Correction
Rate Correction
Rate Correction
Cycle 2n
Cycle 2n+1
Cycle 2 (n+1)
Cycle 2 (n+1) +1
time

Fig. 41

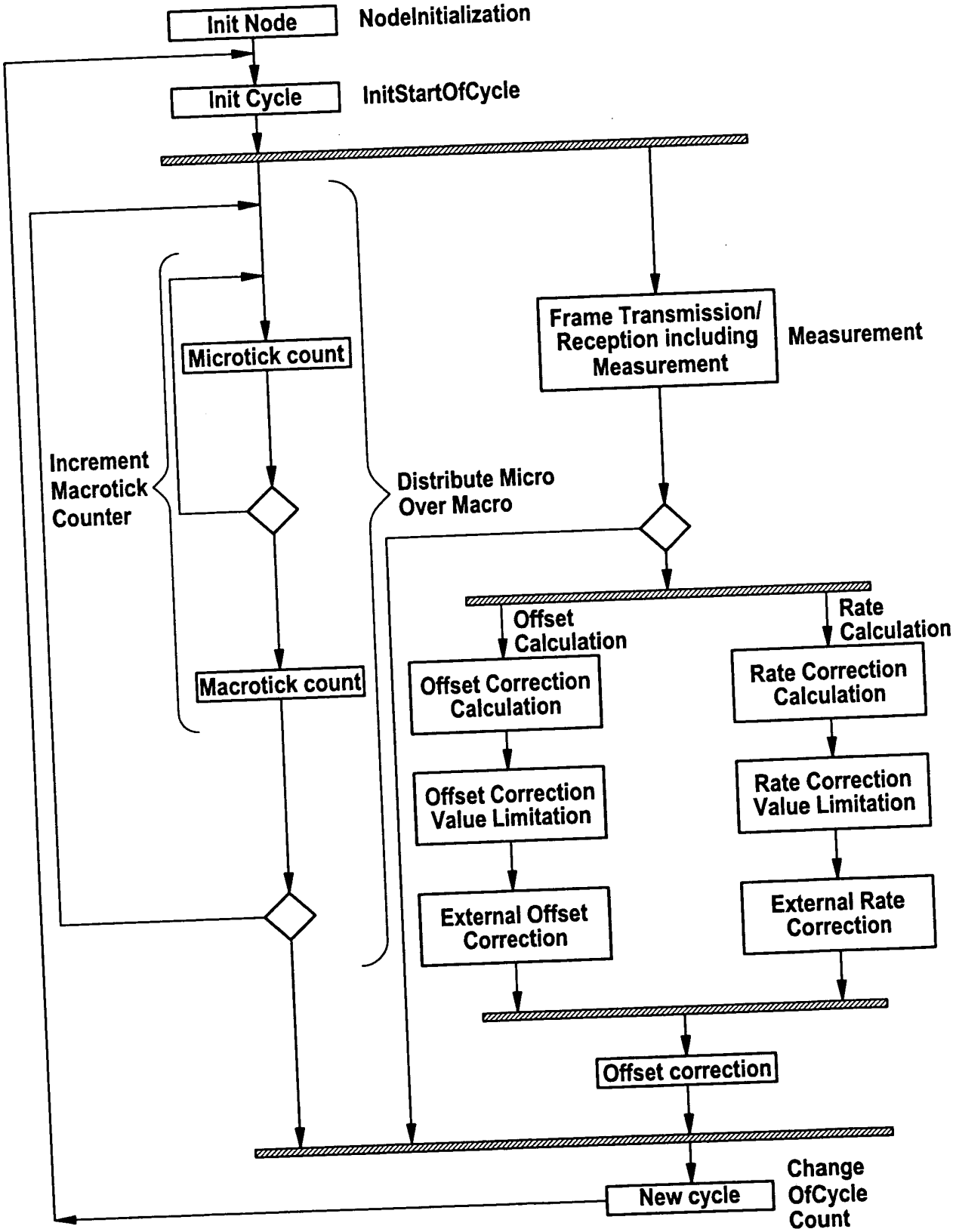

Init Node
NodeInitialization
Init Cycle
InitStartOfCycle
Microtick count
Frame Transmission/ Reception including Measurement
Measurement
Increment Macrotick Counter
Distribute Micro Over Macro
Macrotick count
Offset Calculation
Rate Calculation
Offset Correction Calculation
Rate Correction Calculation
Offset Correction Value Limitation
Rate Correction Value Limitation
External Offset Correction
External Rate Correction
Offset correction
New cycle
Change OfCycle Count

Fig. 42

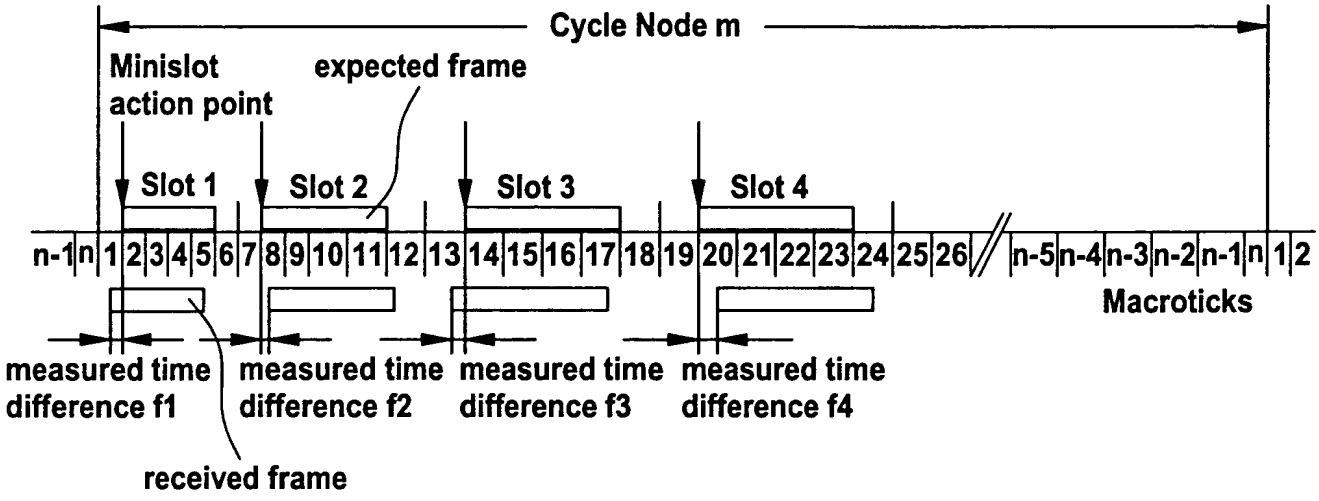


Fig. 43

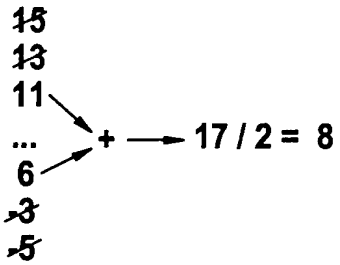


Fig. 44

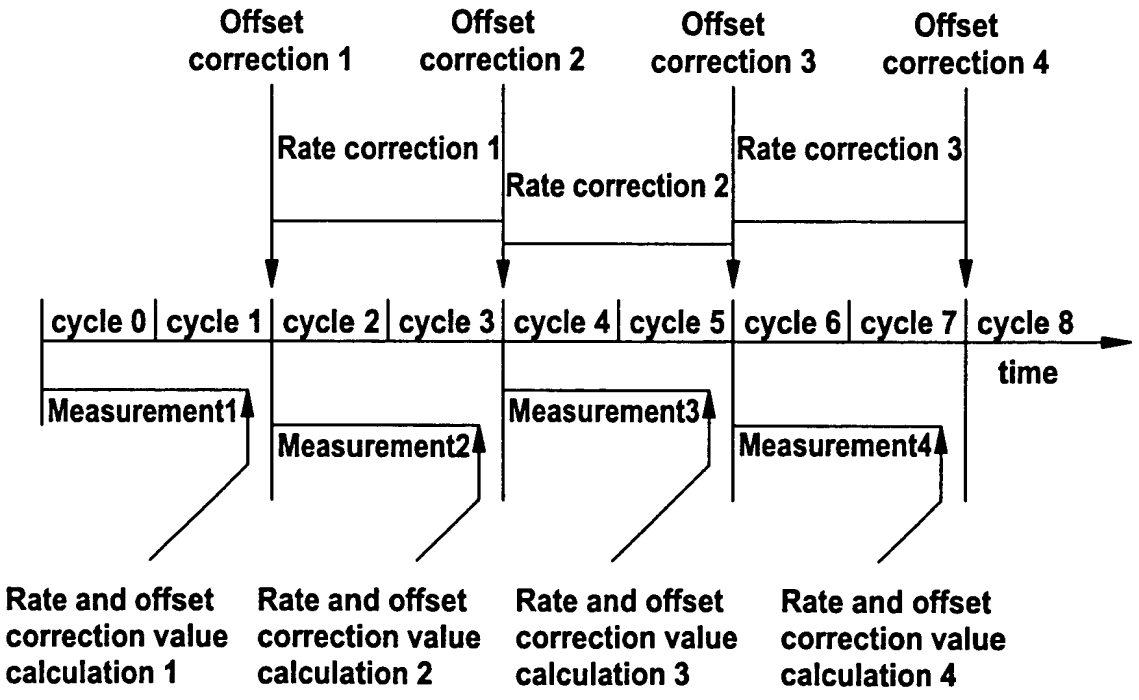


Fig. 45

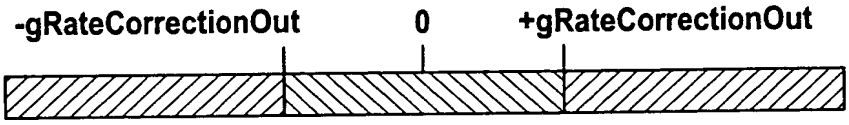


Fig. 46

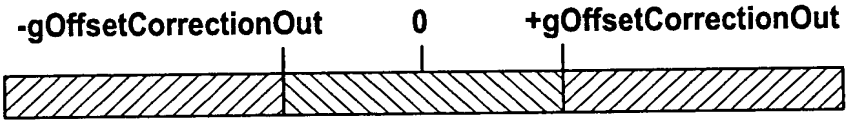


Fig. 47

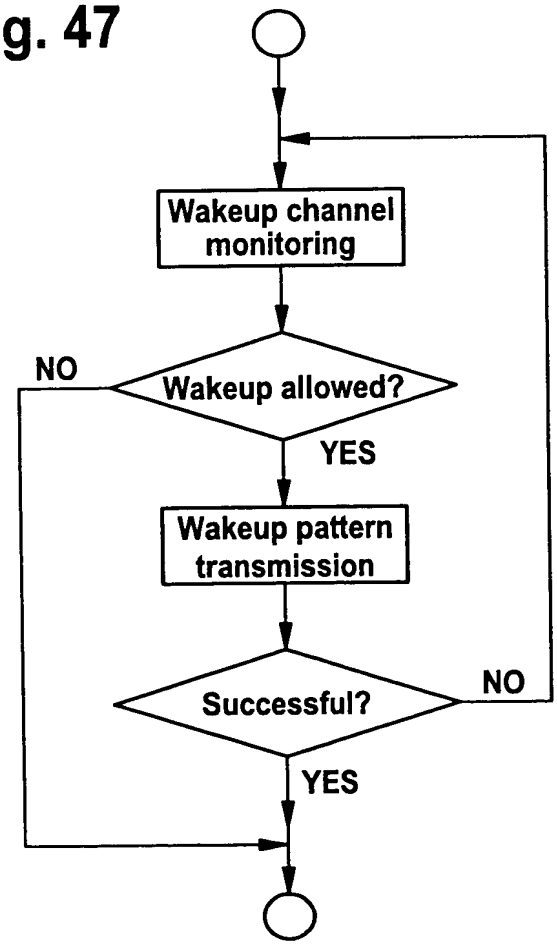


Fig. 48

Wakeup channel monitoring

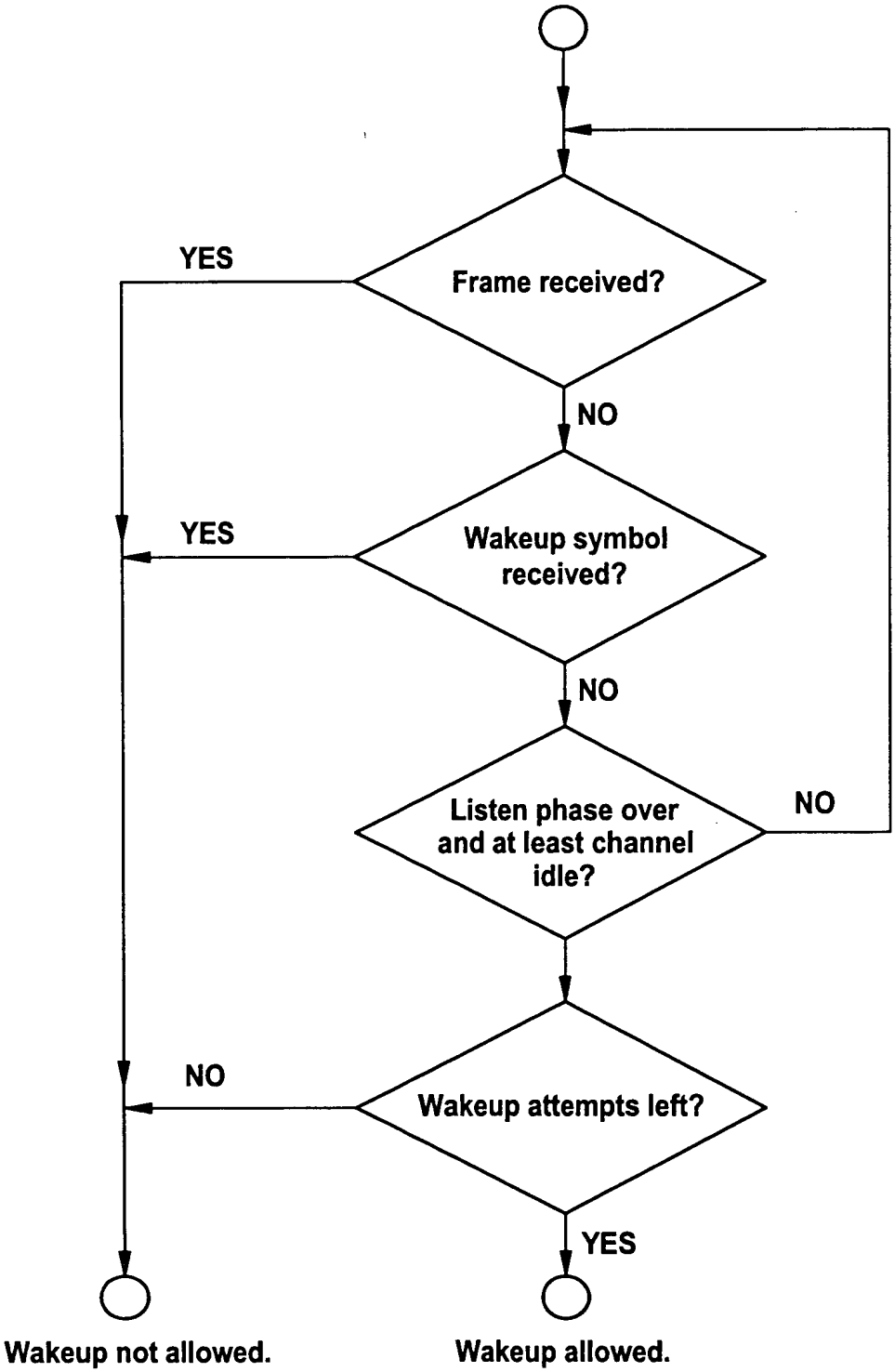


Fig. 49

Wakeup pattern transmission

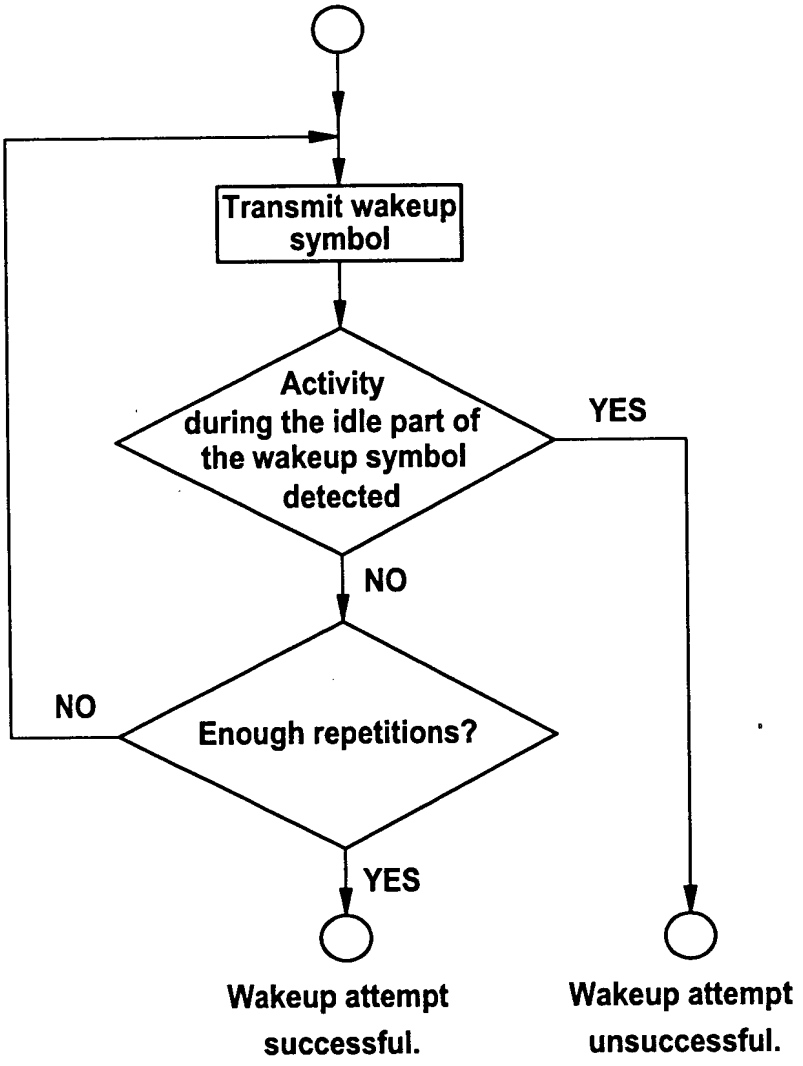


Fig. 50

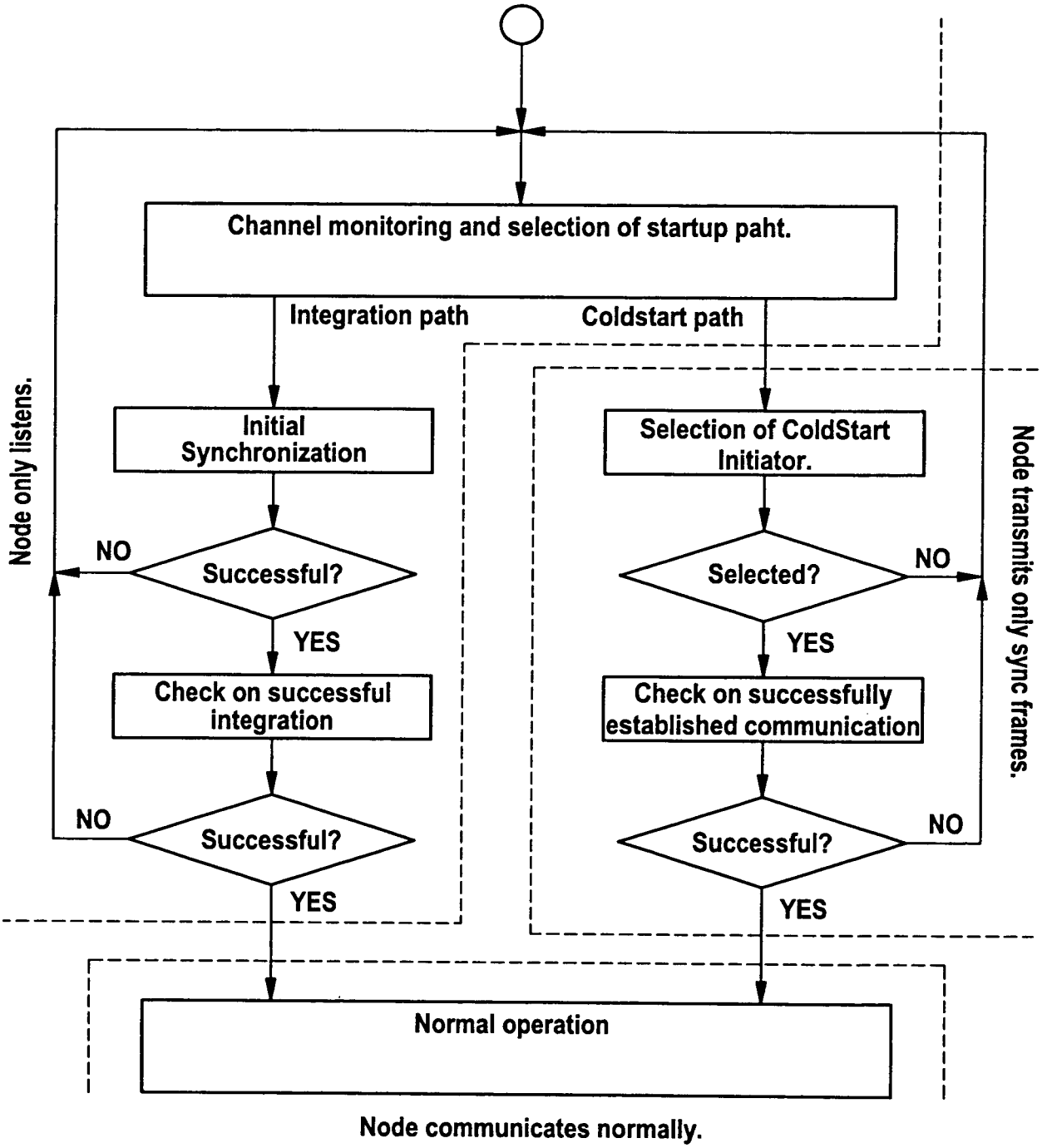

Channel monitoring and selection of startup paht.
Integration path
Coldstart path
Initial Synchronization
Selection of ColdStart Initiator.
Node only listens.
Node transmits only sync frames.
NO
Successful?
YES
Check on successful integration
Selected?
NO
YES
Check on successfully established communication
NO
Successful?
YES
NO
Successful?
YES
Normal operation
Node communicates normally.

Fig. 51

Channel monitoring and selection of startup path.

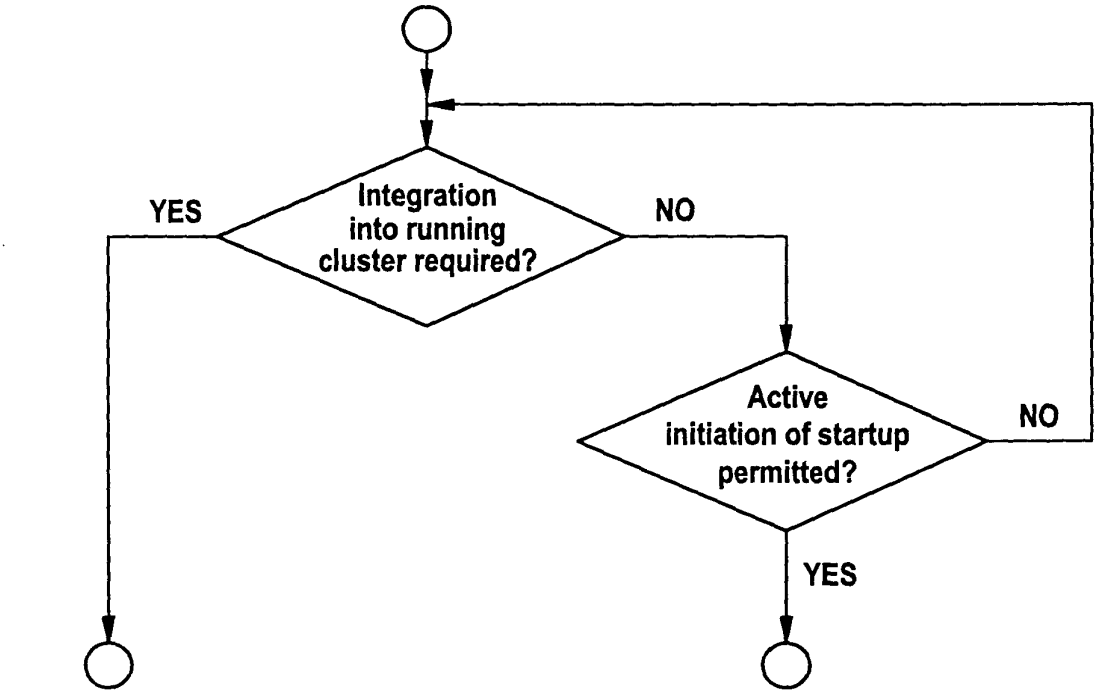


Fig. 52

Selection of ColdStart Initiator

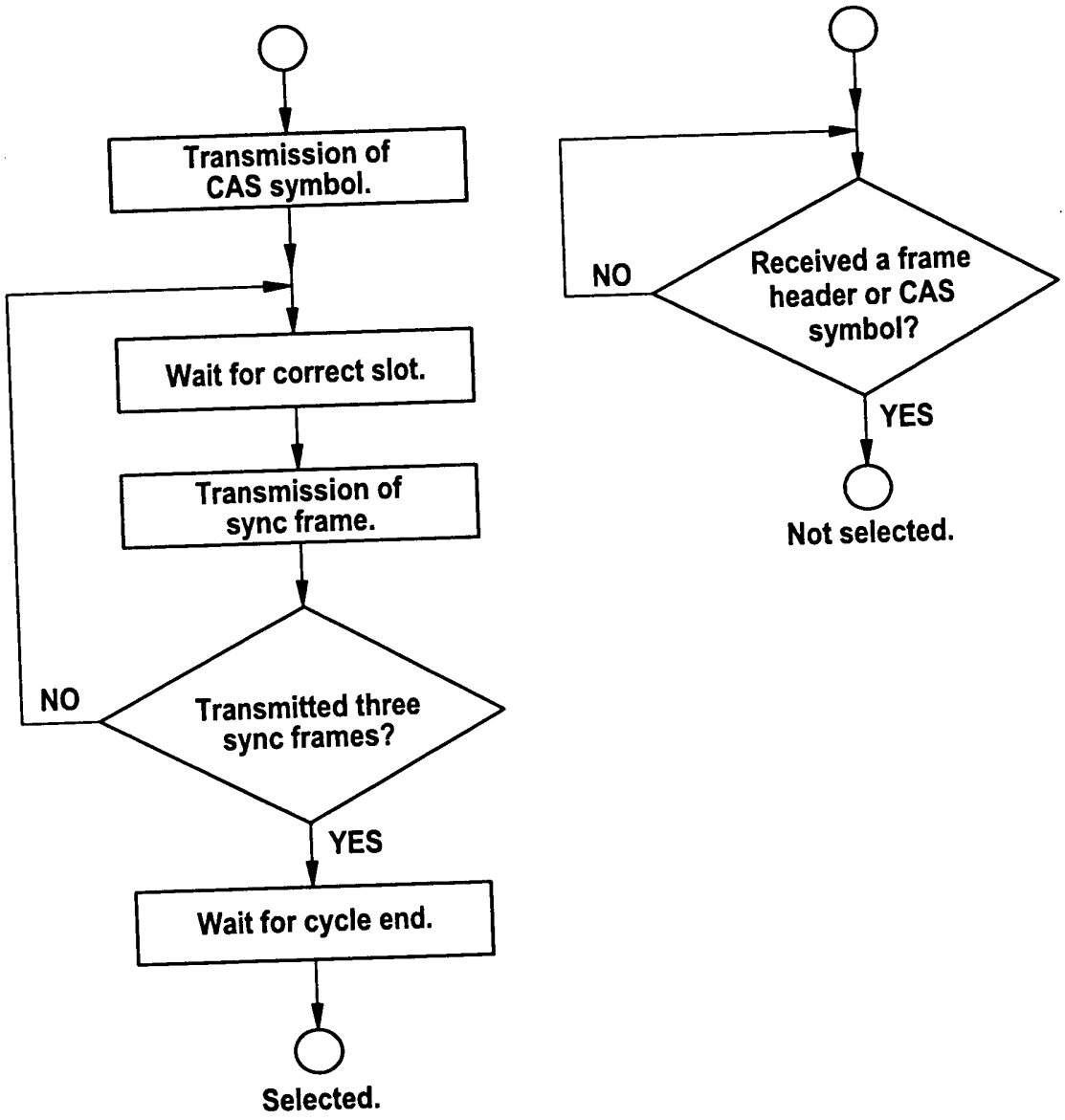


Fig. 53

Check on successfully established communication

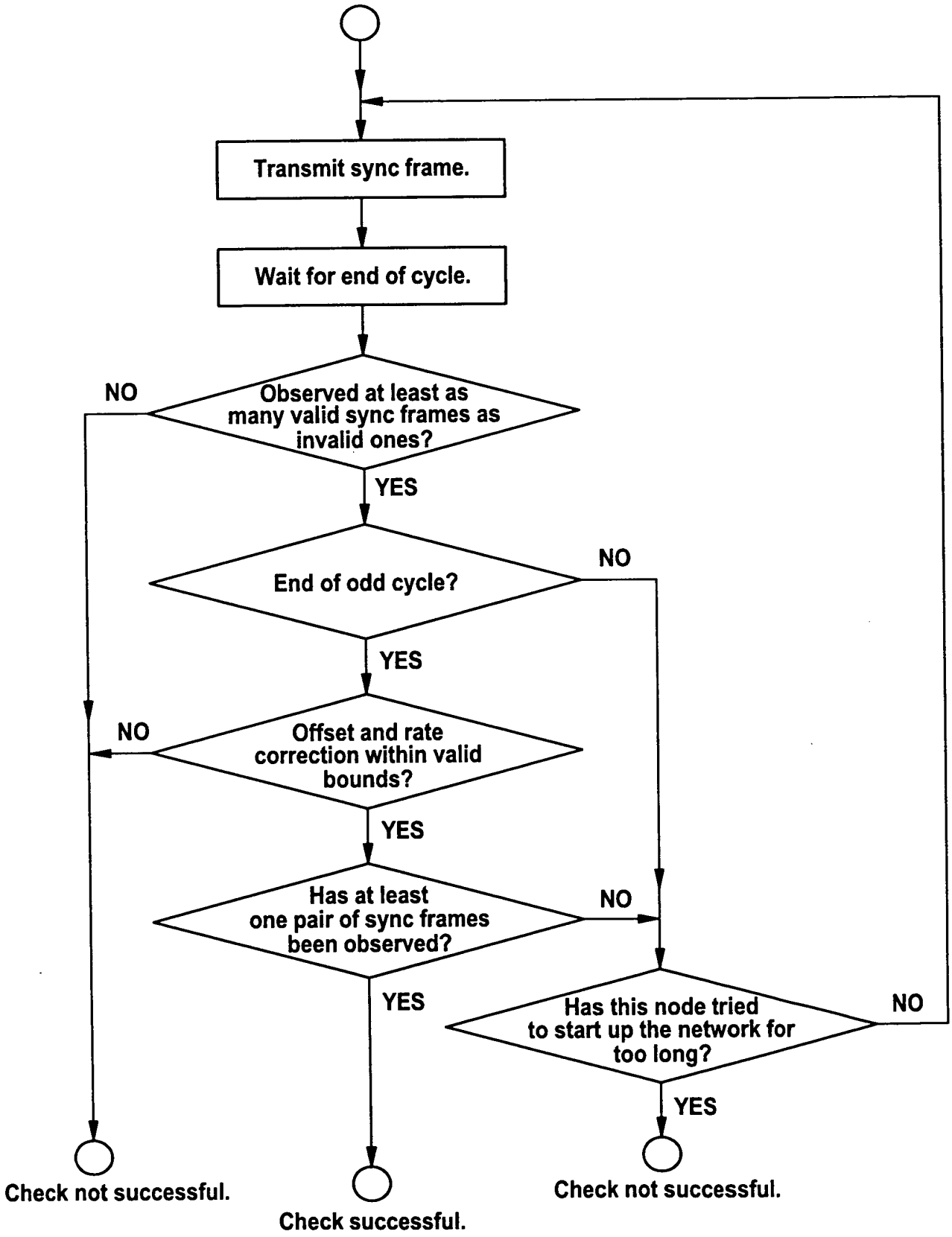


Fig. 54

Initial Synchronization

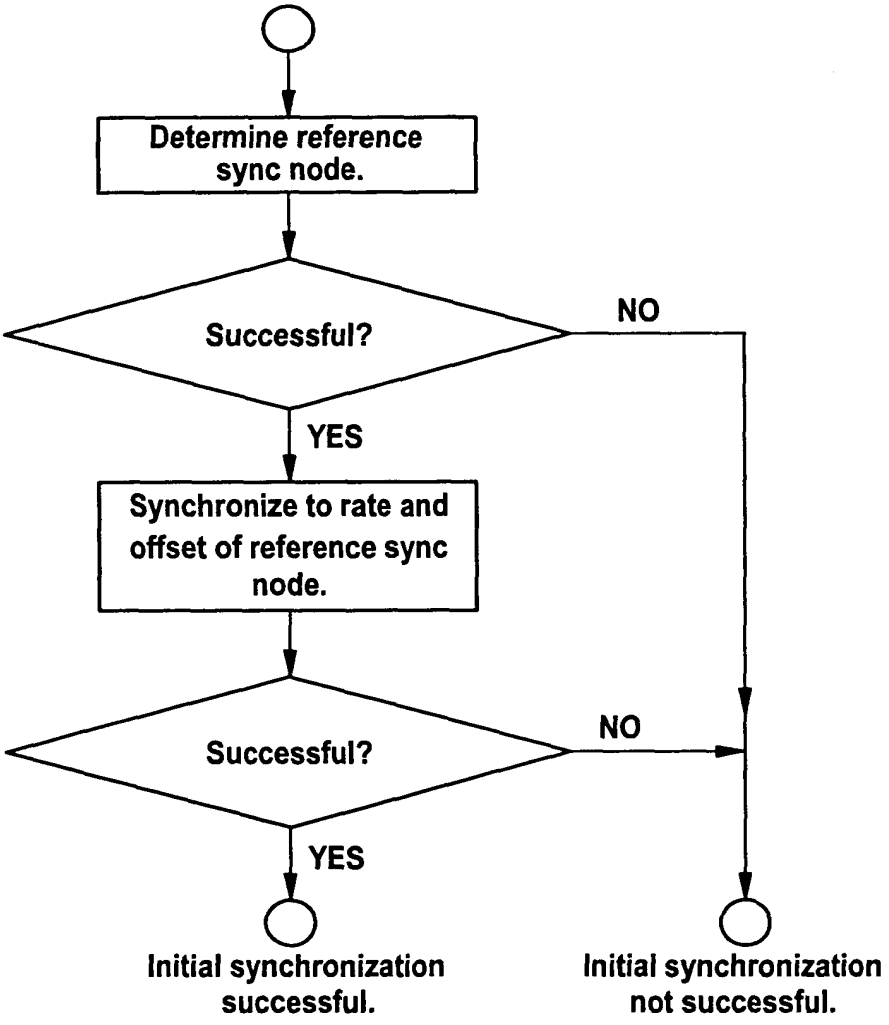


Fig. 55

Check on successful integration

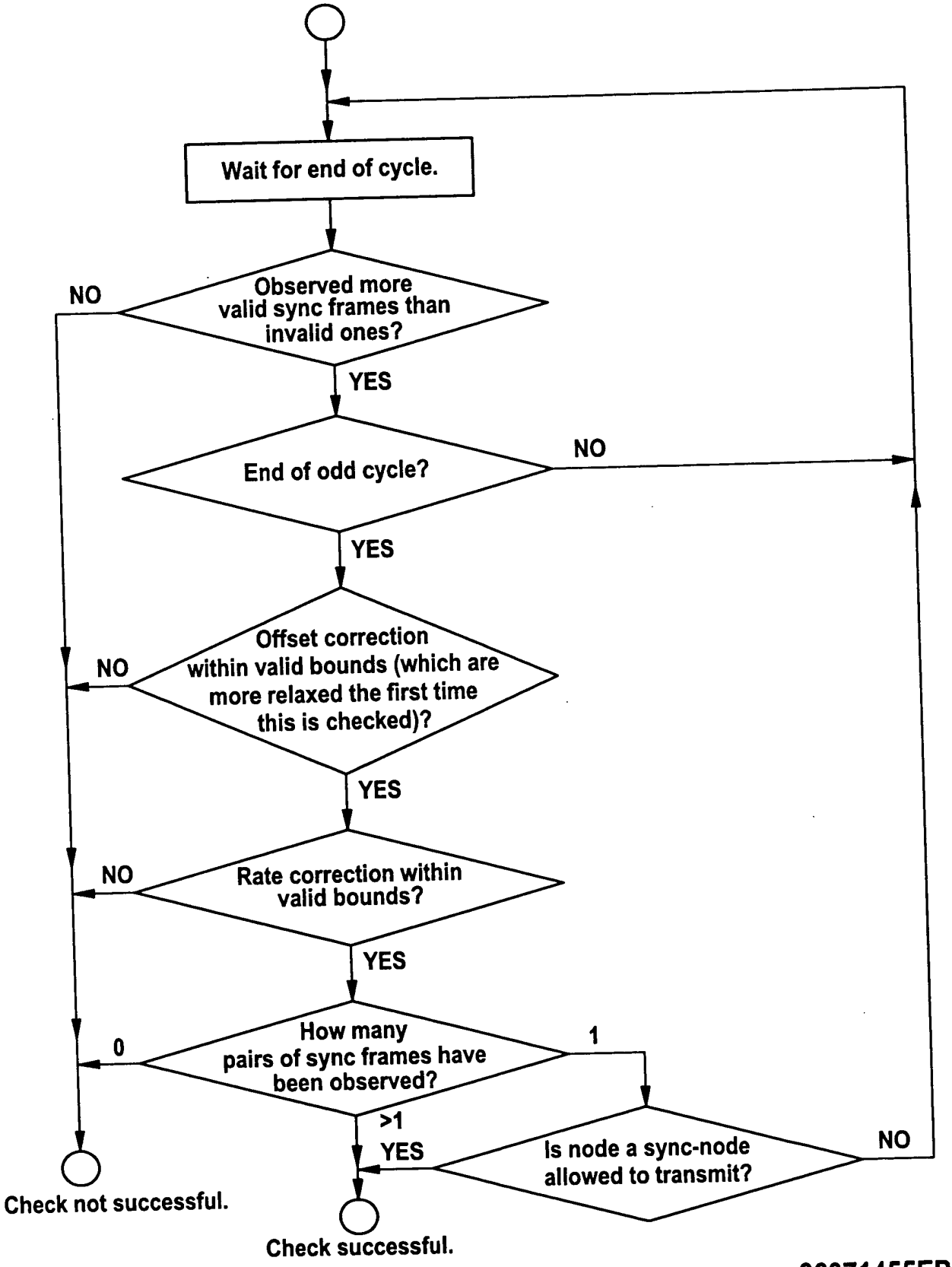


Fig. 56

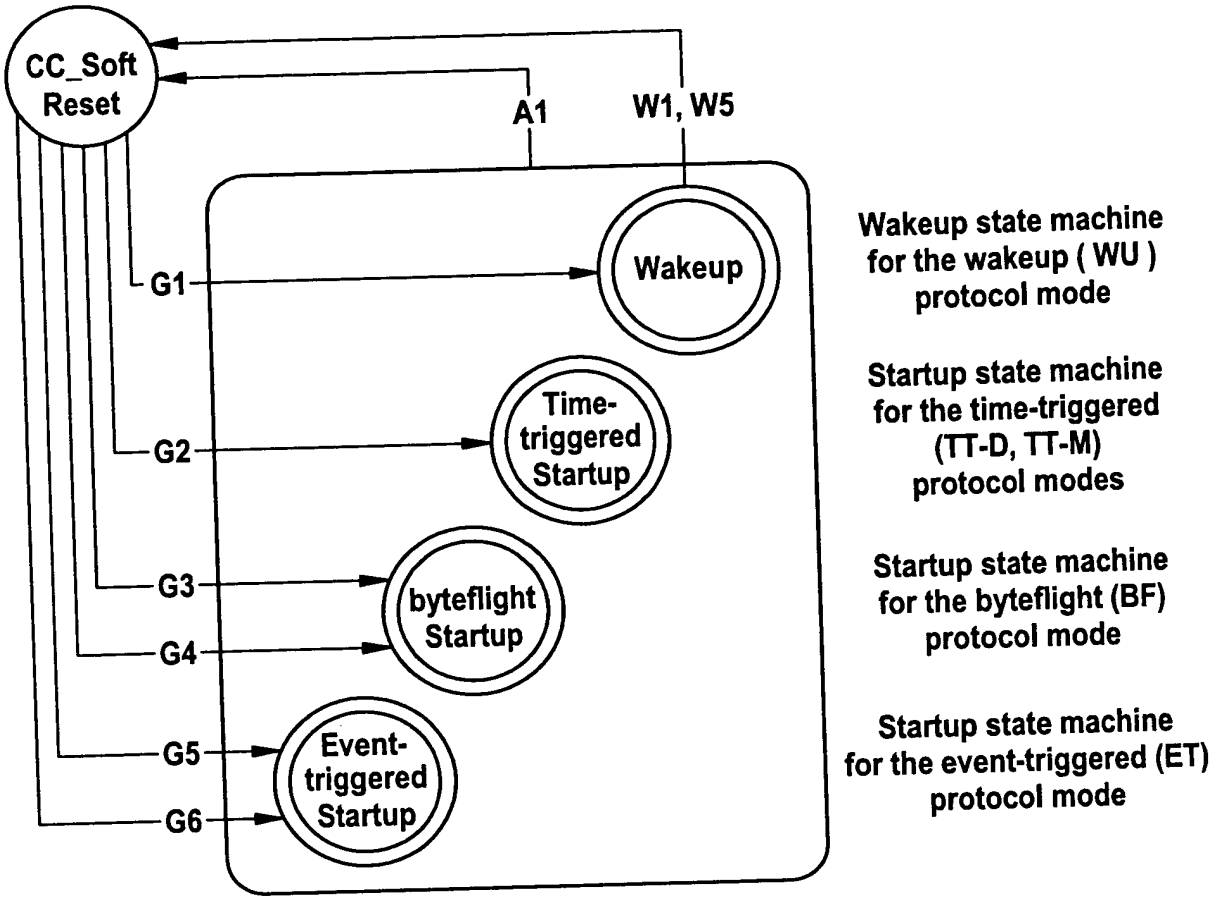

CC_Soft Reset
A1
W1, W5
G1
Wakeup
G2
Time-triggered Startup
G3
G4
byteflight Startup
G5
G6
Event-triggered Startup
Wakeup state machine for the wakeup (WU) protocol mode
Startup state machine for the time-triggered (TT-D, TT-M) protocol modes
Startup state machine for the byteflight (BF) protocol mode
Startup state machine for the event-triggered (ET) protocol mode

Fig. 57

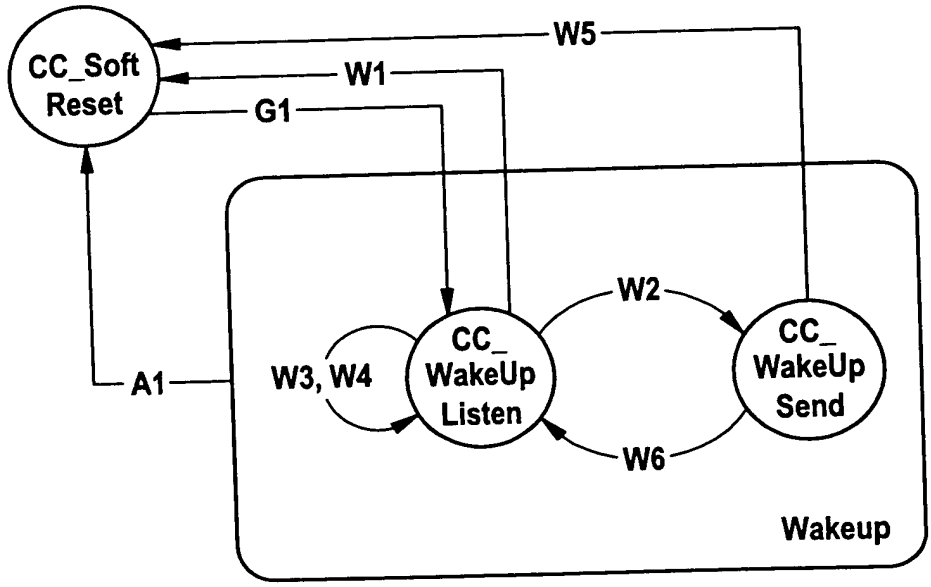

CC_Soft Reset
W5
W1
G1
A1
W3, W4
CC_ WakeUp Listen
W2
CC_ WakeUp Send
W6
Wakeup

Fig. 58

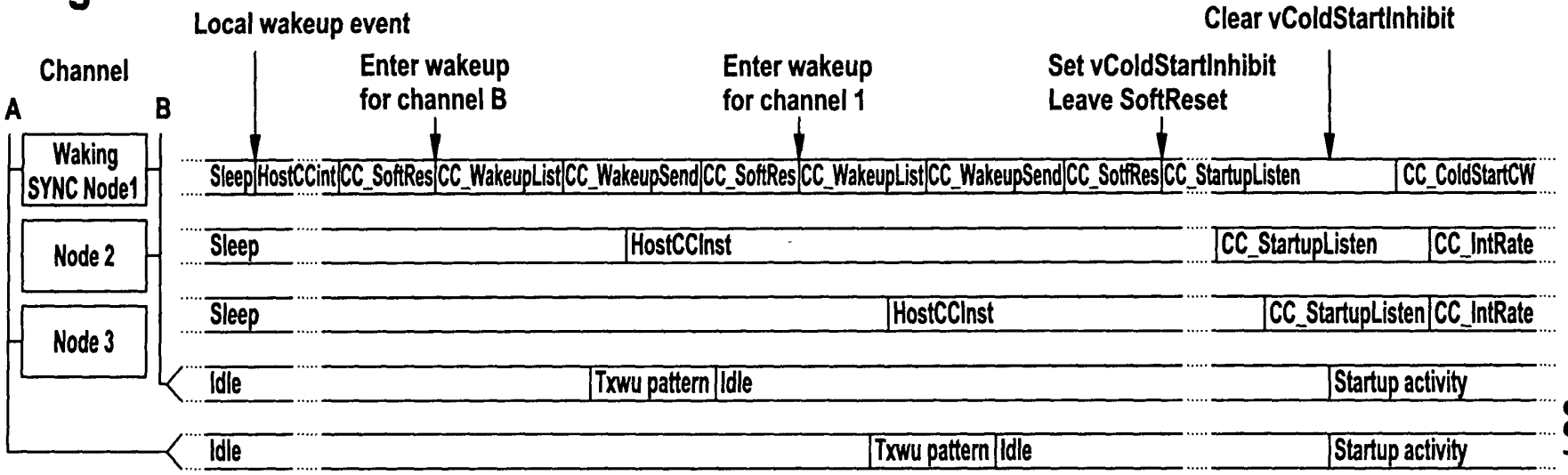

Local wakeup event
Clear vColdStartInhibit
Channel
Enter wakeup for channel B
Enter wakeup for channel 1
Set vColdStartInhibit Leave SoftReset
A
B
Waking SYNC Node1
Node 2
Node 3
Sleep
HostCCint
CC_SoftRes
CC_WakeupList
CC_WakeupSend
CC_SoftRes
CC_WakeupList
CC_WakeupSend
CC_SofRes
CC_StartupListen
CC_ColdStartCW
Sleep
HostCCInst
CC_StartupListen
CC_IntRate
Sleep
HostCCInst
CC_StartupListen
CC_IntRate
Idle
Txwu pattern
Idle
Startup activity
Idle
Txwu pattern
Idle
Startup activity

Fig. 59

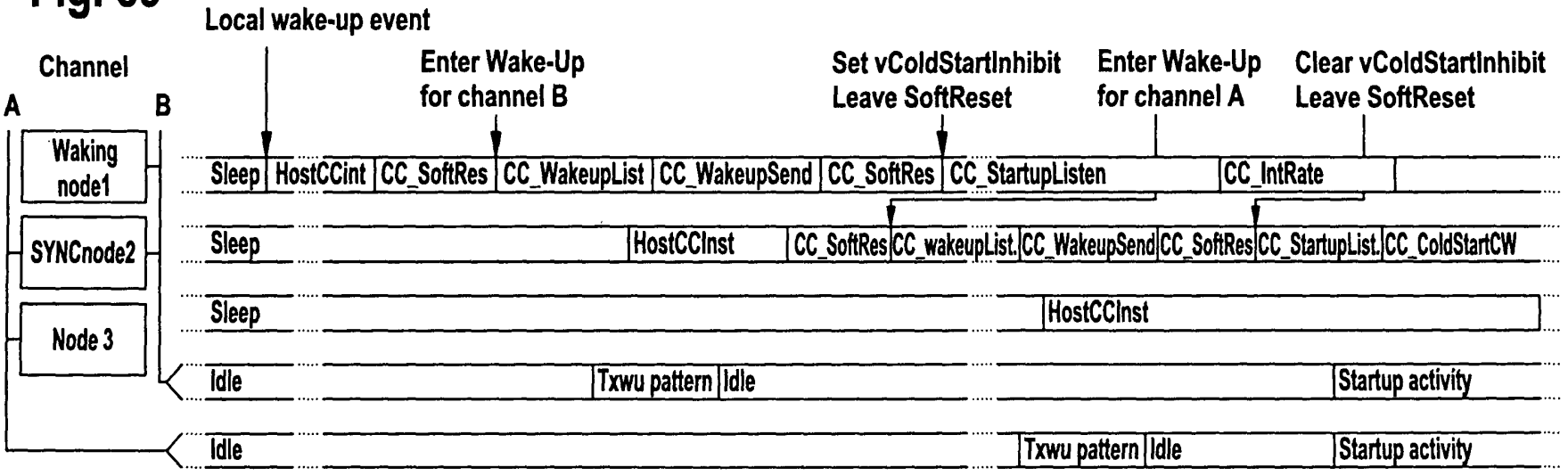

Local wake-up event
Channel
Enter Wake-Up for channel B
Set vColdStartInhibit Leave SoftReset
Enter Wake-Up for channel A
Clear vColdStartInhibit Leave SoftReset
A
B
Waking node1
SYNCnode2
Node 3
Sleep
HostCCint
CC_SoftRes
CC_WakeupList
CC_WakeupSend
CC_SoftRes
CC_StartupListen
CC_IntRate
Sleep
HostCCInst
CC_SoftRes
CC_wakeupList.
CC_WakeupSend
CC_SoftRes
CC_StartupList.
CC_ColdStartCW
Sleep
HostCCInst
Idle
Txwu pattern
Idle
Startup activity
Idle
Txwu pattern
Idle
Startup activity

Fig. 60

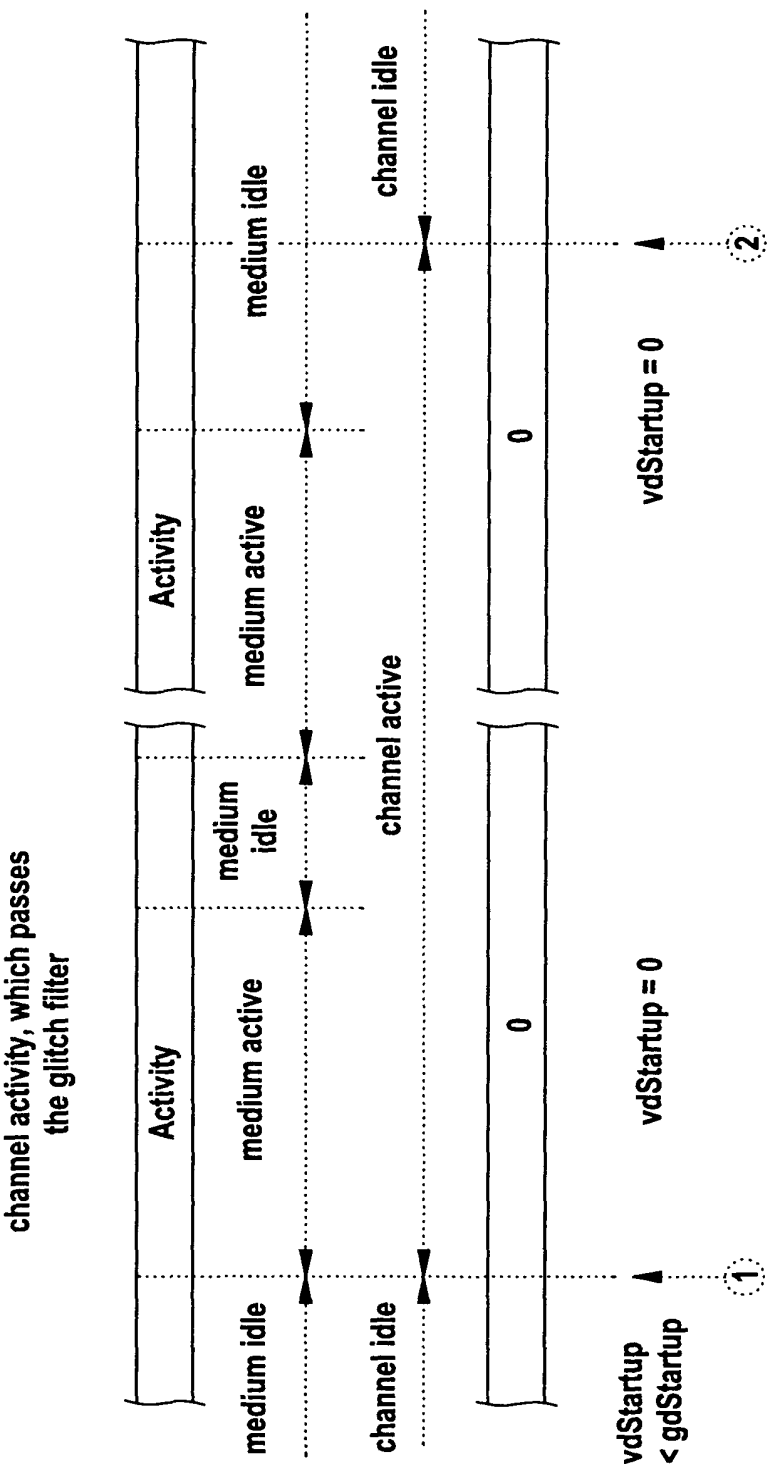

channel activity, which passes the glitch filter
Activity
medium active
medium idle
channel active
channel idle
0
vdStartup = 0
vdStartup < gdStartup
1
2

Fig. 61

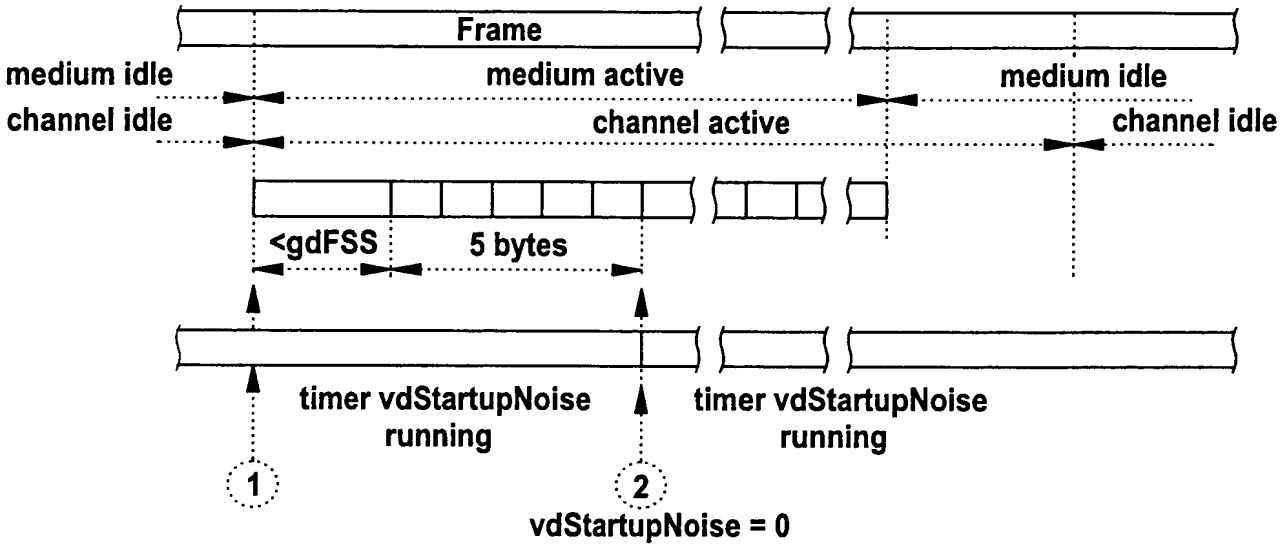

Frame
medium idle
medium active
medium idle
channel idle
channel active
channel idle
<gdFSS
5 bytes
timer vdStartupNoise running
timer vdStartupNoise running
1
2
vdStartupNoise = 0

Fig. 62

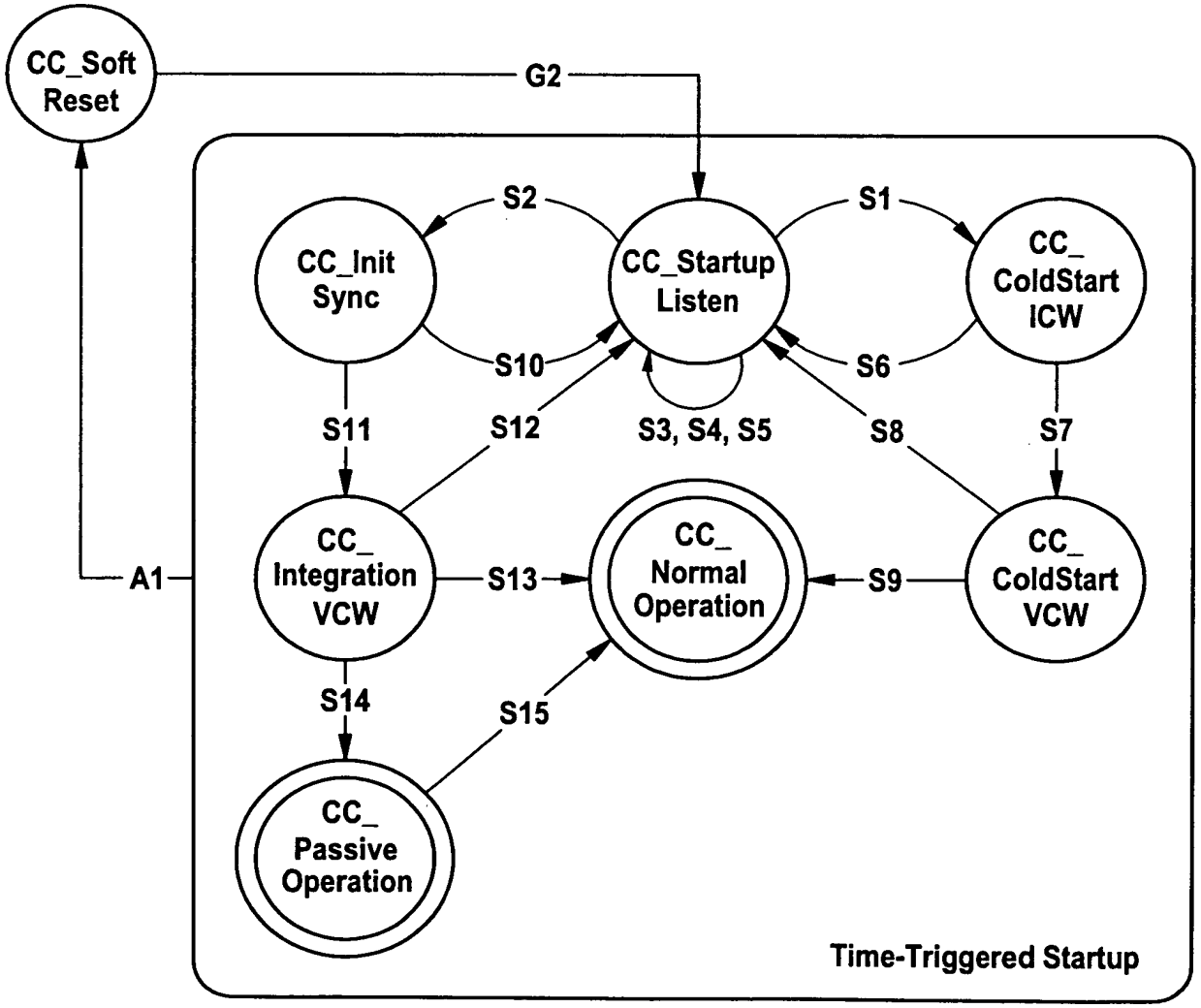

CC_Soft Reset
G2
CC_Init Sync
CC_Startup Listen
CC_ ColdStart ICW
S2
S1
S10
S6
S11
S12
S3, S4, S5
S8
S7
CC_ Integration VCW
CC_ Normal Operation
CC_ ColdStart VCW
A1
S13
S9
S14
S15
CC_ Passive Operation
Time-Triggered Startup

Fig. 63

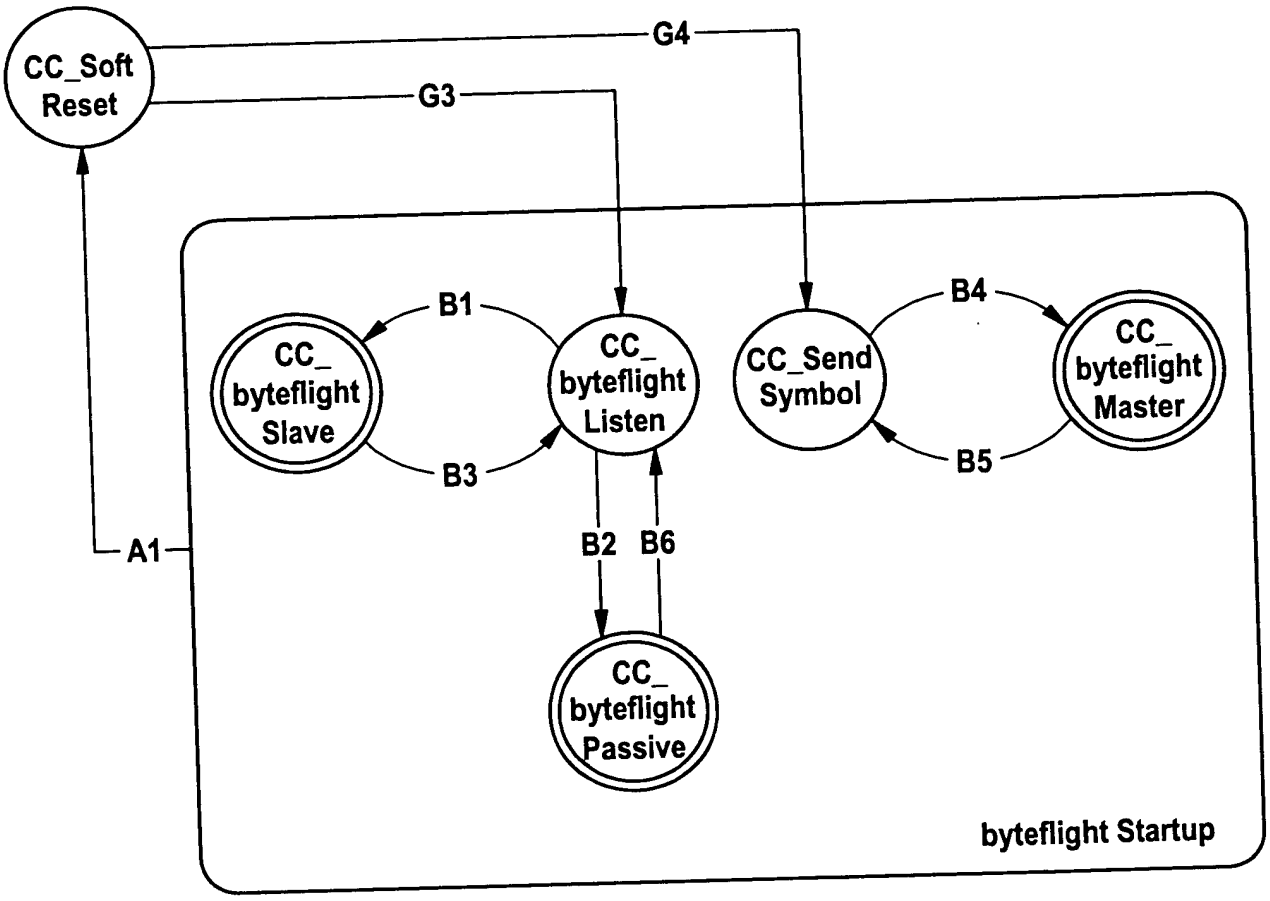

CC_Soft Reset
G4
G3
A1
B1
B3
B2
B6
B4
B5
CC_ byteflight Slave
CC_ byteflight Listen
CC_Send Symbol
CC_ byteflight Master
CC_ byteflight Passive
byteflight Startup

Fig. 64

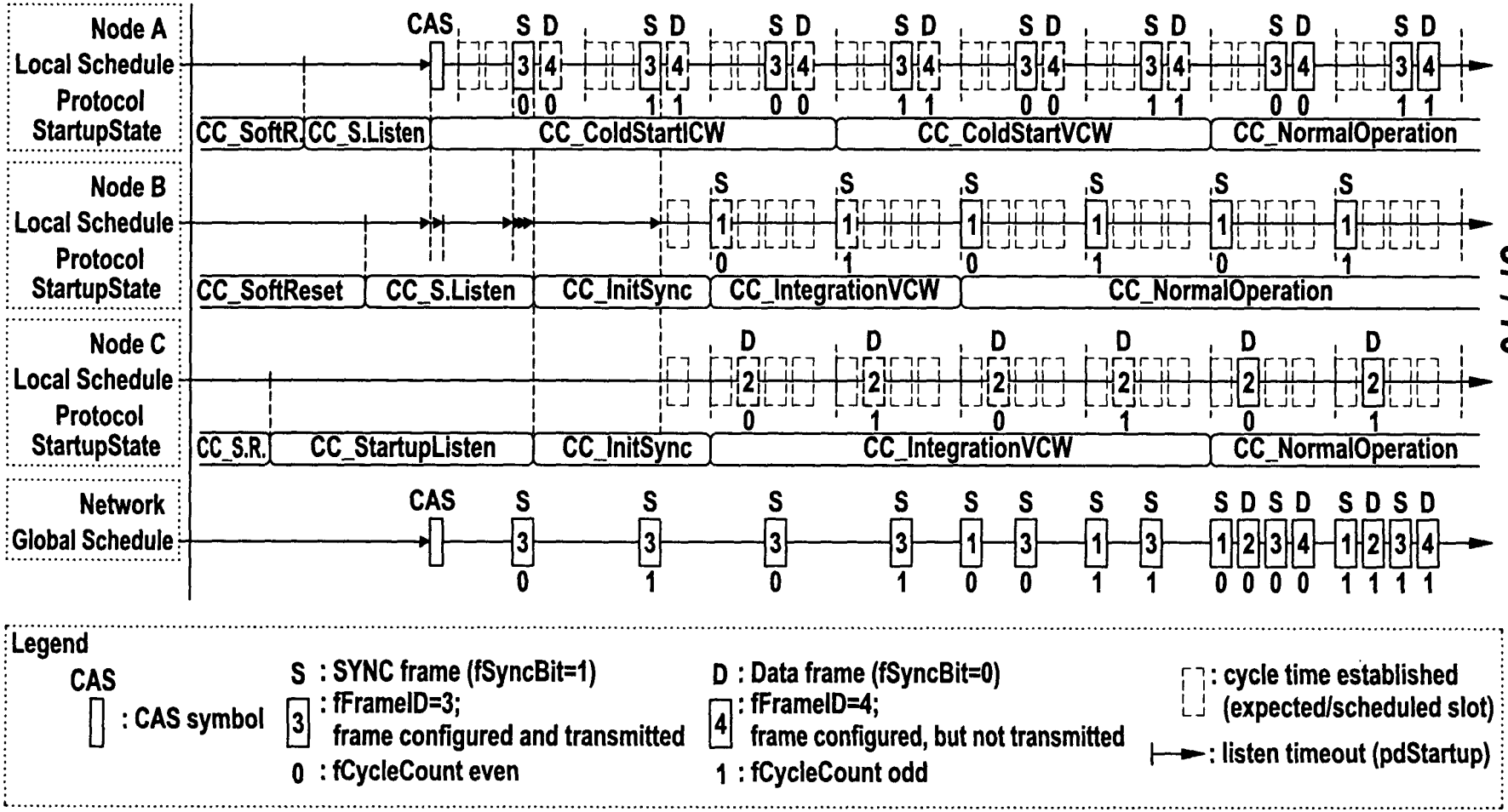

Node A
Local Schedule
Protocol StartupState
CAS
S D
CC_SoftR
CC_S.Listen
CC_ColdStartICW
CC_ColdStartVCW
CC_NormalOperation
Node B
Local Schedule
Protocol StartupState
S
CC_SoftReset
CC_S.Listen
CC_InitSync
CC_IntegrationVCW
CC_NormalOperation
Node C
Local Schedule
Protocol StartupState
D
CC_S.R.
CC_StartupListen
CC_InitSync
CC_IntegrationVCW
CC_NormalOperation
Network
Global Schedule
CAS
S
D
Legend
CAS
: CAS symbol
S : SYNC frame (fSyncBit=1)
3 : fFrameID=3; frame configured and transmitted
0 : fCycleCount even
D : Data frame (fSyncBit=0)
4 : fFrameID=4; frame configured, but not transmitted
1 : fCycleCount odd
: cycle time established (expected/scheduled slot)
: listen timeout (pdStartup)

Fig. 65

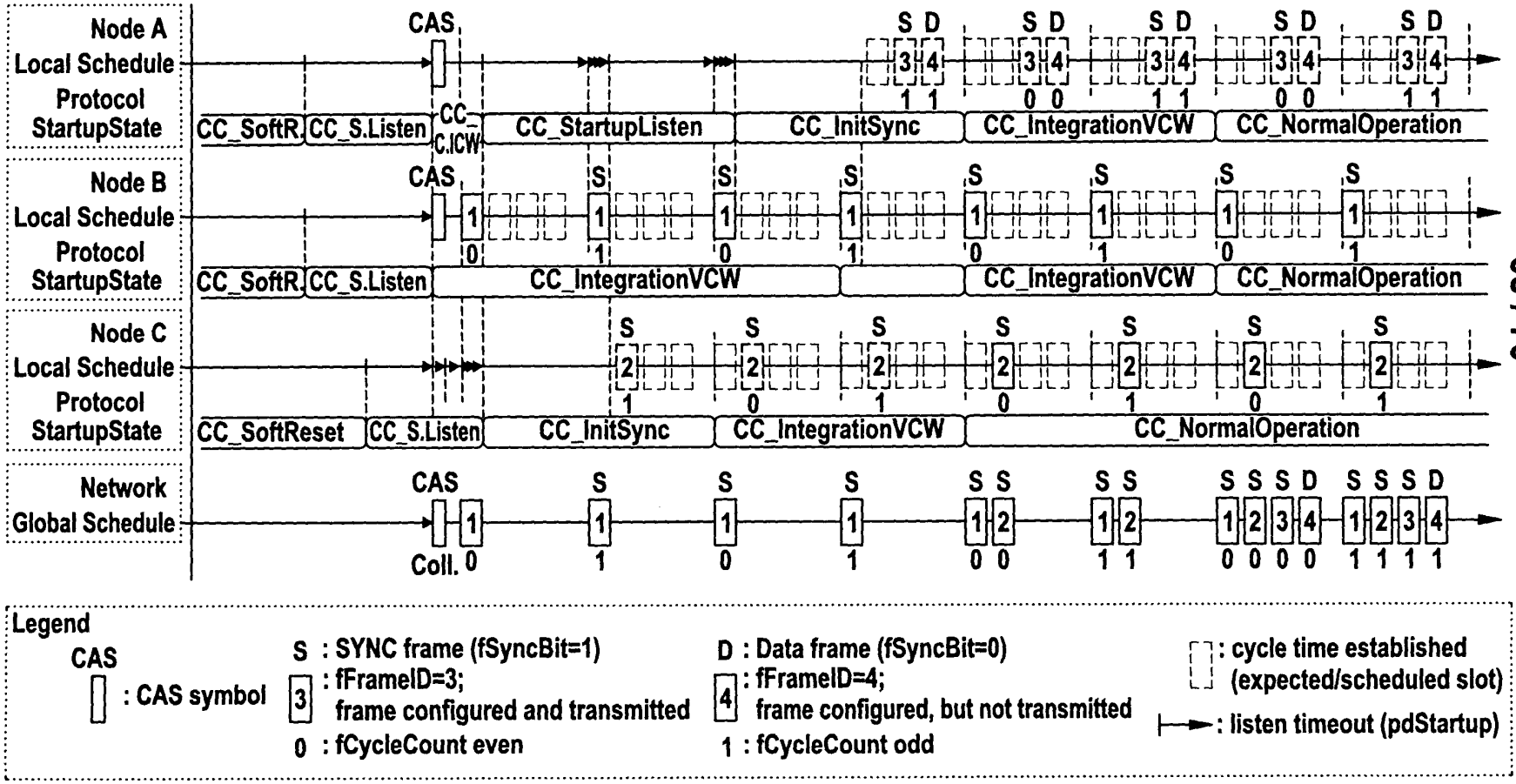
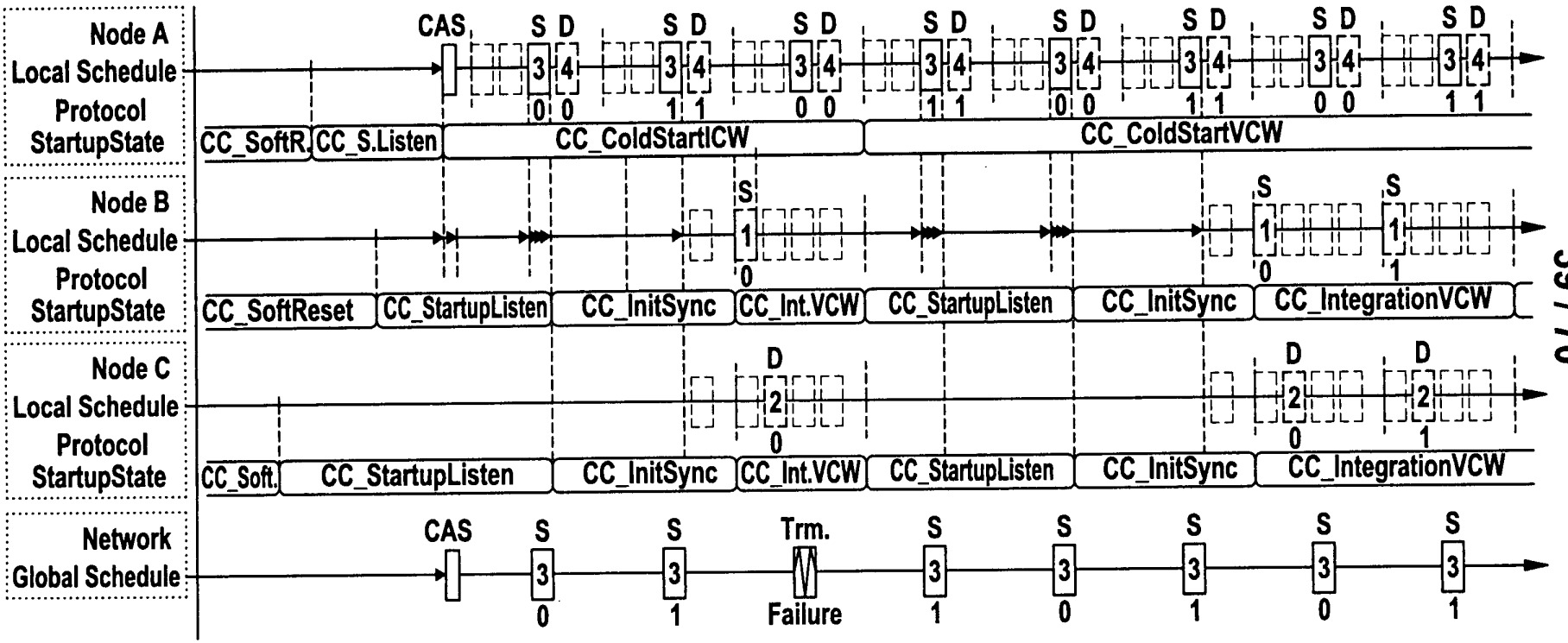

Node A
Local Schedule
Protocol StartupState
CAS
S D
3 4
1 1
0 0
CC_SoftR
CC_S.Listen
CC_C.ICW
CC_StartupListen
CC_InitSync
CC_IntegrationVCW
CC_NormalOperation
Node B
Local Schedule
Protocol StartupState
CAS
S
1
0
CC_SoftR
CC_S.Listen
CC_IntegrationVCW
CC_IntegrationVCW
CC_NormalOperation
Node C
Local Schedule
Protocol StartupState
S
2
1
0
CC_SoftReset
CC_S.Listen
CC_InitSync
CC_IntegrationVCW
CC_NormalOperation
Network
Global Schedule
CAS
S
S S
S S D
S S S D
1
2
3
4
Coll. 0
0 0
1 1
0 0 0 0
1 1 1 1
Legend
CAS
: CAS symbol
S : SYNC frame (fSyncBit=1)
3 : fFrameID=3; frame configured and transmitted
0 : fCycleCount even
D : Data frame (fSyncBit=0)
4 : fFrameID=4; frame configured, but not transmitted
1 : fCycleCount odd
: cycle time established (expected/scheduled slot)
: listen timeout (pdStartup)

Fig. 66



Legend

CAS : CAS symbol

S : SYNC frame (fSyncBit=1)

3 : fFrameID=3; frame configured and transmitted

0 : fCycleCount even

D : Data frame (fSyncBit=0)

4 : fFrameID=4; frame configured, but not transmitted

1 : fCycleCount odd

: cycle time established (expected/scheduled slot)

: listen timeout (pdStartup)

Fig. 67

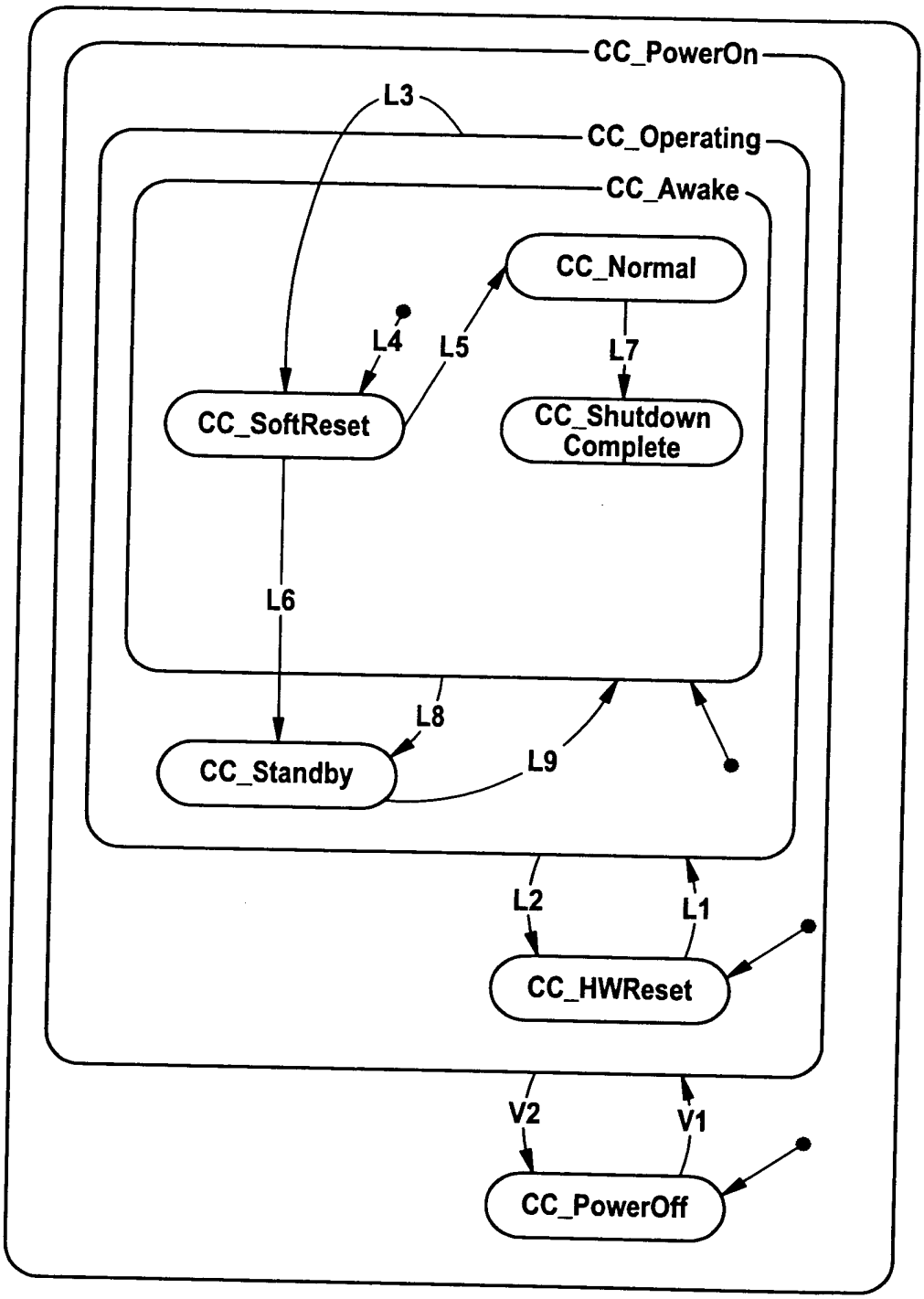

CC_PowerOn
L3
CC_Operating
CC_Awake
CC_Normal
L4
L5
L7
CC_SoftReset
CC_Shutdown Complete
L6
L8
CC_Standby
L9
L2
L1
CC_HWReset
V2
V1
CC_PowerOff

Fig. 68

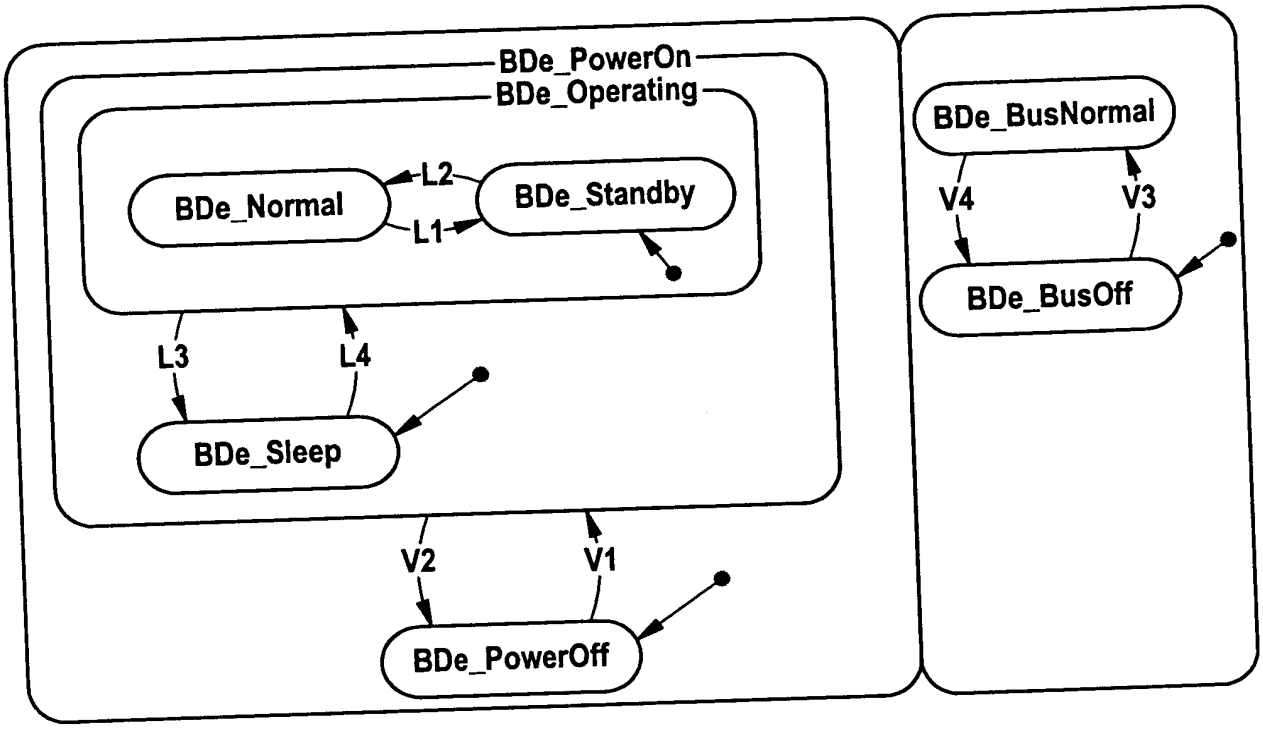

BDe_PowerOn
BDe_Operating
BDe_Normal
L2
L1
BDe_Standby
L3
L4
BDe_Sleep
V2
V1
BDe_PowerOff
BDe_BusNormal
V4
V3
BDe_BusOff

Fig. 69

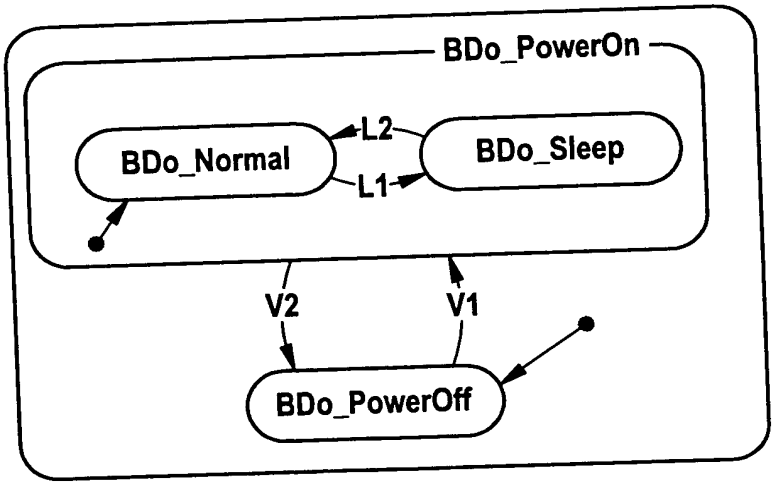

BDo_PowerOn
BDo_Normal
L2
L1
BDo_Sleep
V2
V1
BDo_PowerOff

Fig. 70

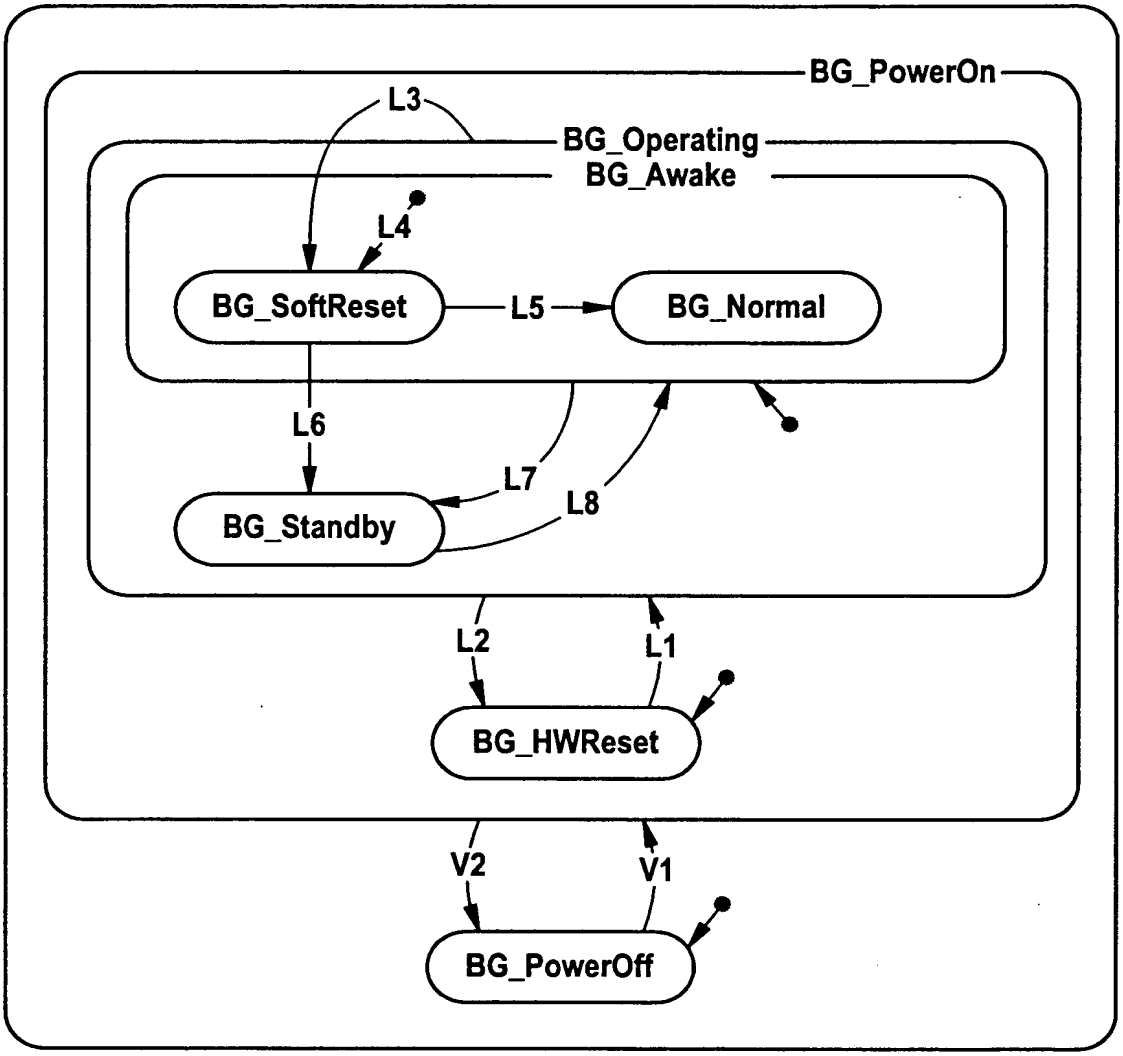

BG_PowerOn
L3
BG_Operating
BG_Awake
L4
BG_SoftReset
L5
BG_Normal
L6
L7
L8
BG_Standby
L2
L1
BG_HWReset
V2
V1
BG_PowerOff

Fig. 71

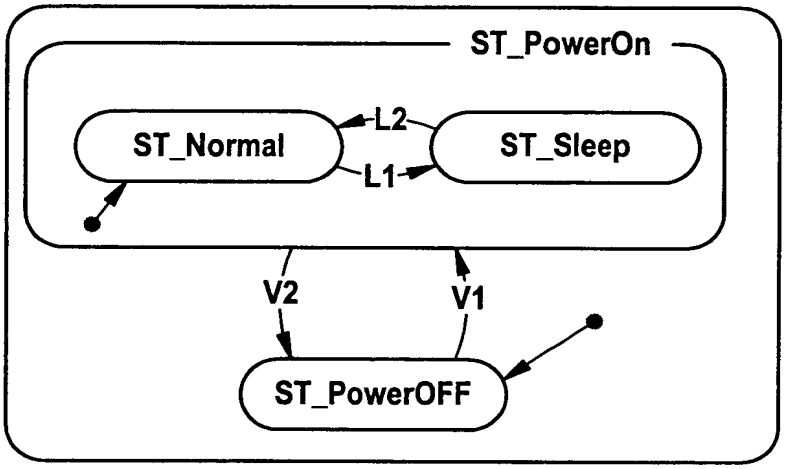


Fig. 72

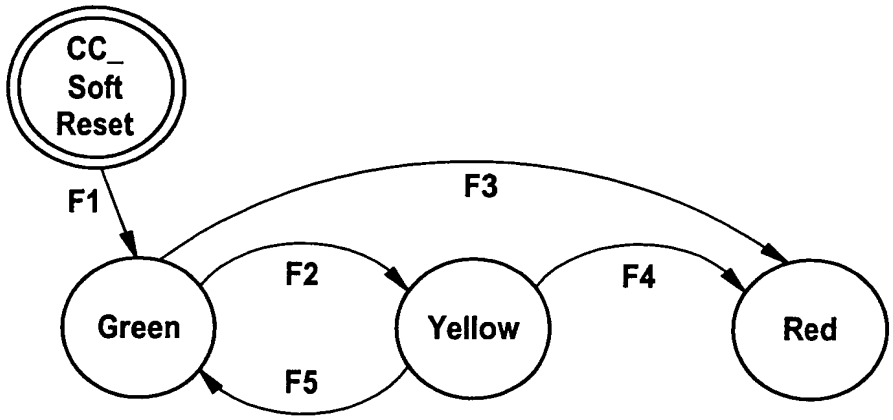


Fig. 73

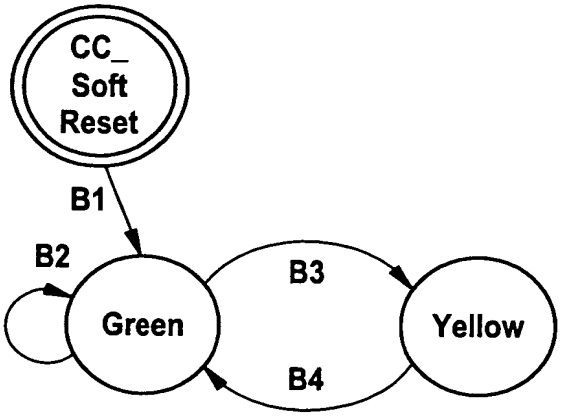


Fig. 74

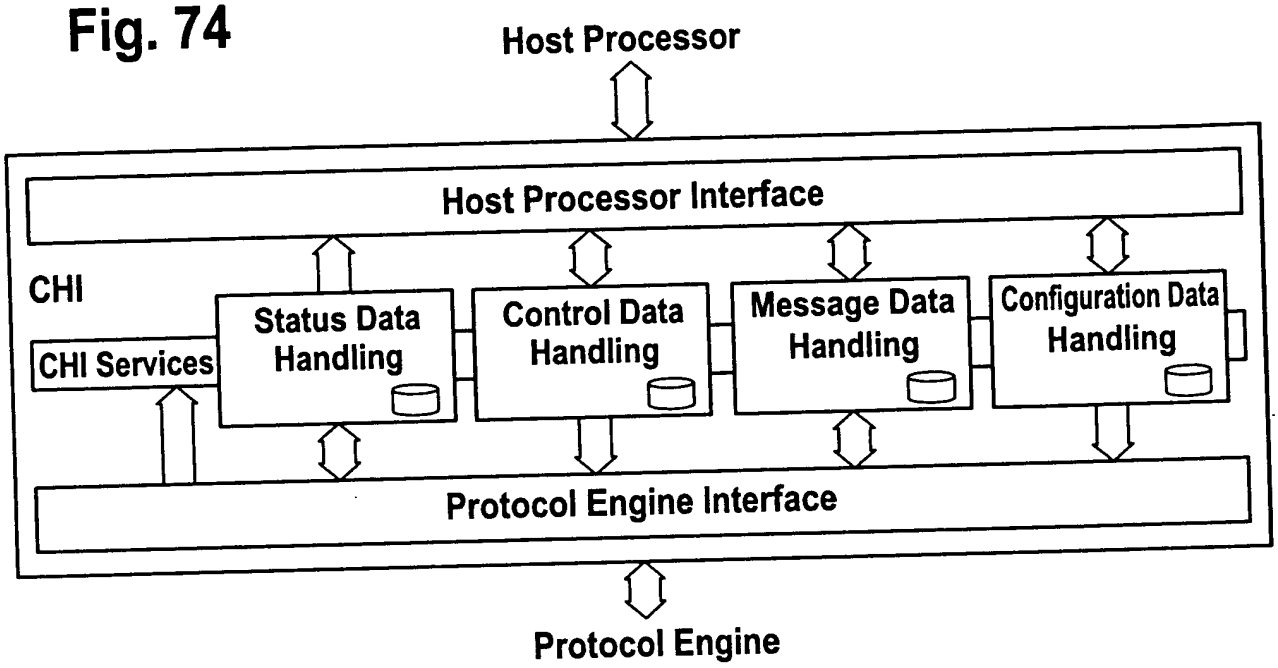

Host Processor
Host Processor Interface
CHI
CHI Services
Status Data Handling
Control Data Handling
Message Data Handling
Configuration Data Handling
Protocol Engine Interface
Protocol Engine

Fig. 75

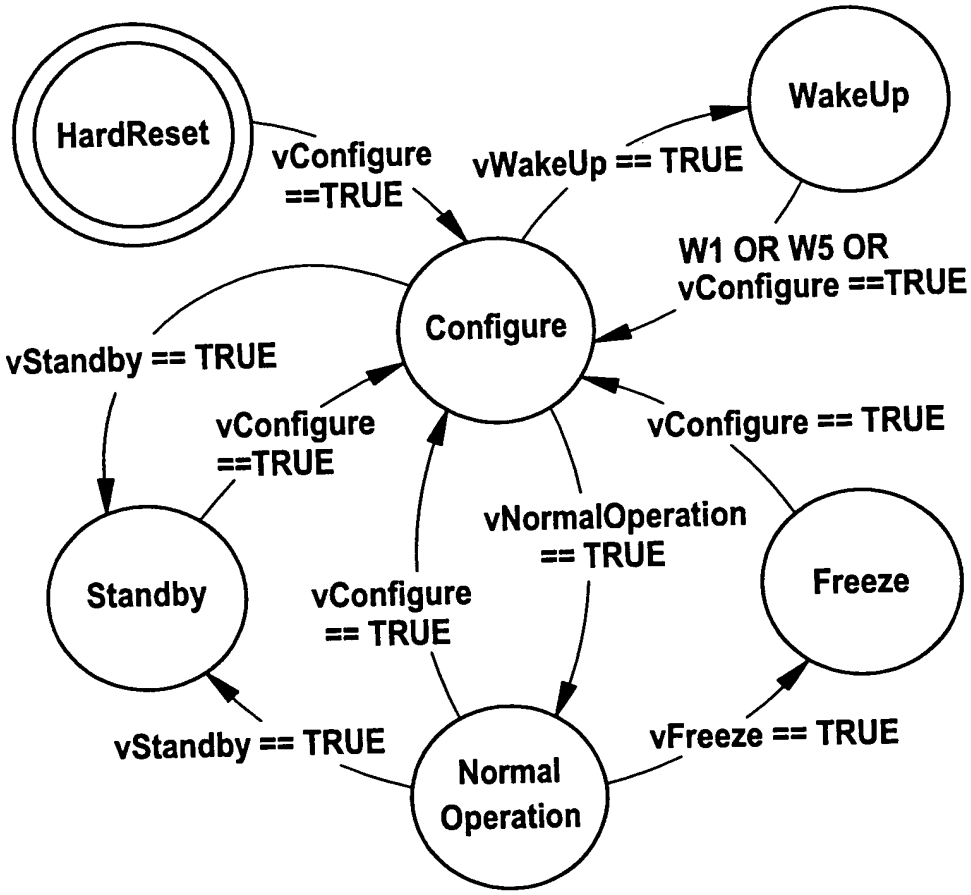

HardReset
vConfigure ==TRUE
WakeUp
vWakeUp == TRUE
W1 OR W5 OR vConfigure ==TRUE
Configure
vStandby == TRUE
vConfigure ==TRUE
vConfigure == TRUE
Standby
vNormalOperation == TRUE
Freeze
vConfigure == TRUE
vStandby == TRUE
Normal Operation
vFreeze == TRUE

Fig. 76

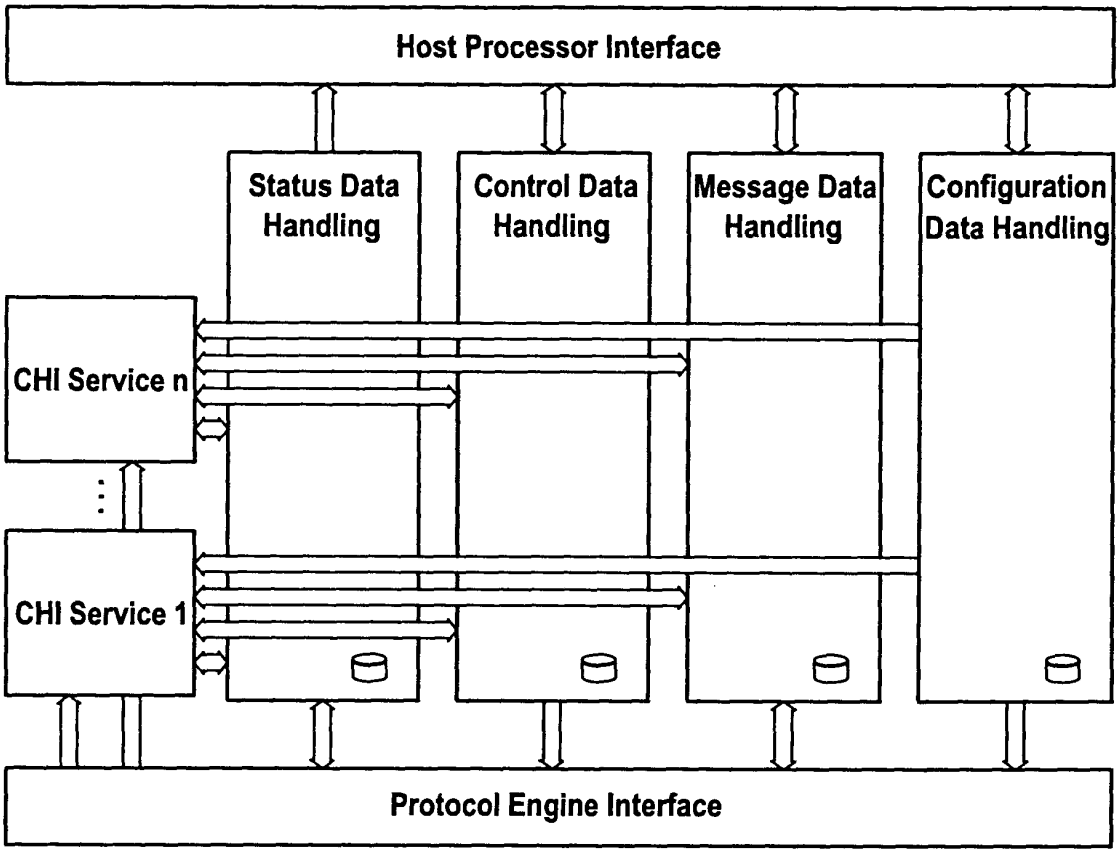

Host Processor Interface
Status Data Handling
Control Data Handling
Message Data Handling
Configuration Data Handling
CHI Service n
CHI Service 1
Protocol Engine Interface

Fig. 77

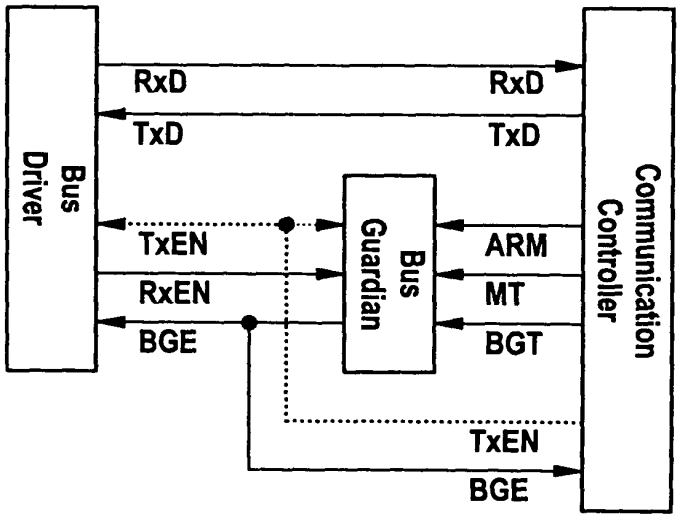

Bus Driver
RxD
TxD
TxEN
RxEN
BGE
Bus Guardian
RxD
TxD
ARM
MT
BGT
TxEN
BGE
Communication Controller

Fig. 78

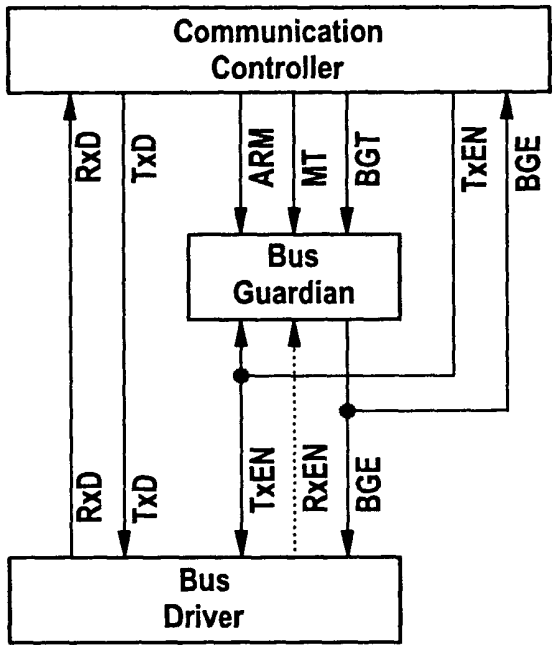


Fig. 79

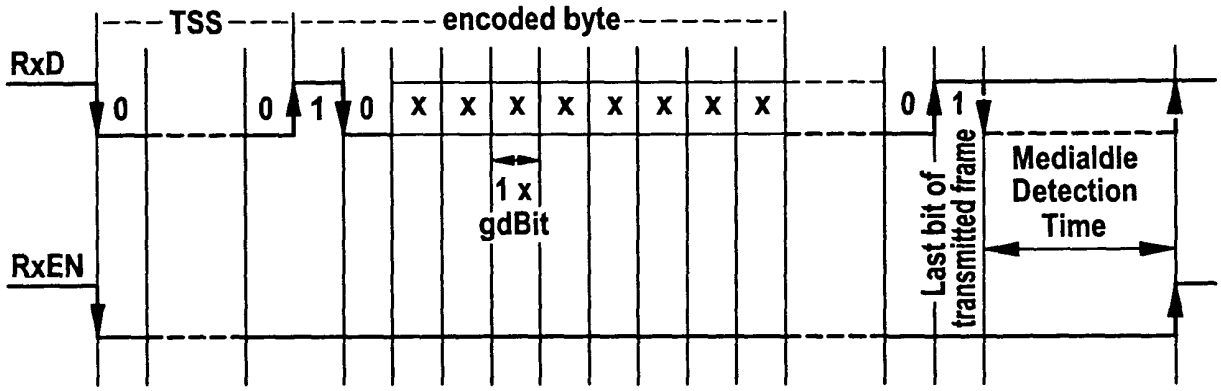


Fig. 80

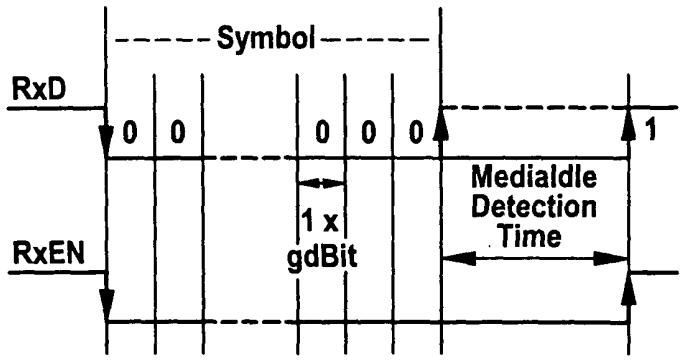


Fig. 81

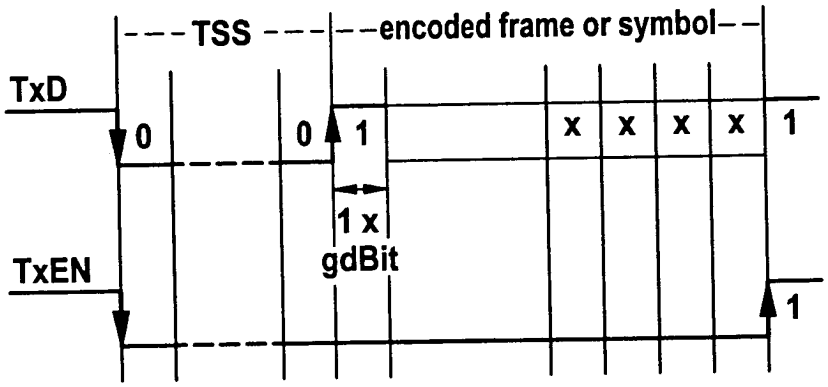

TSS
encoded frame or symbol
TxD
0
0
1
x
x
x
x
1
1 x
gdBit
TxEN
1

Fig. 82

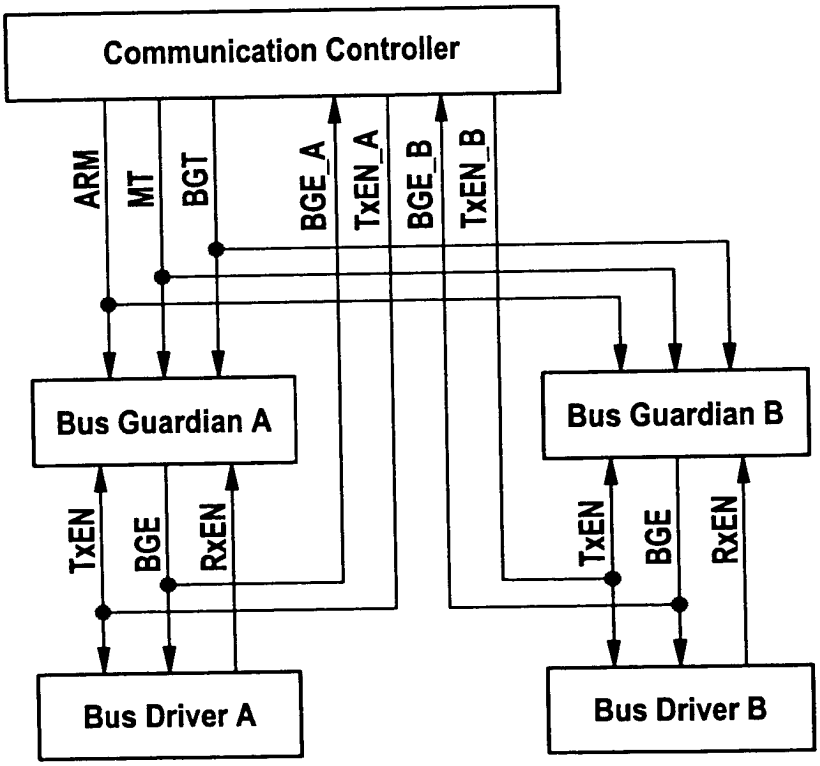

Communication Controller
ARM
MT
BGT
BGE_A
TxEN_A
BGE_B
TxEN_B
Bus Guardian A
Bus Guardian B
TxEN
BGE
RxEN
TxEN
BGE
RxEN
Bus Driver A
Bus Driver B

Fig. 83

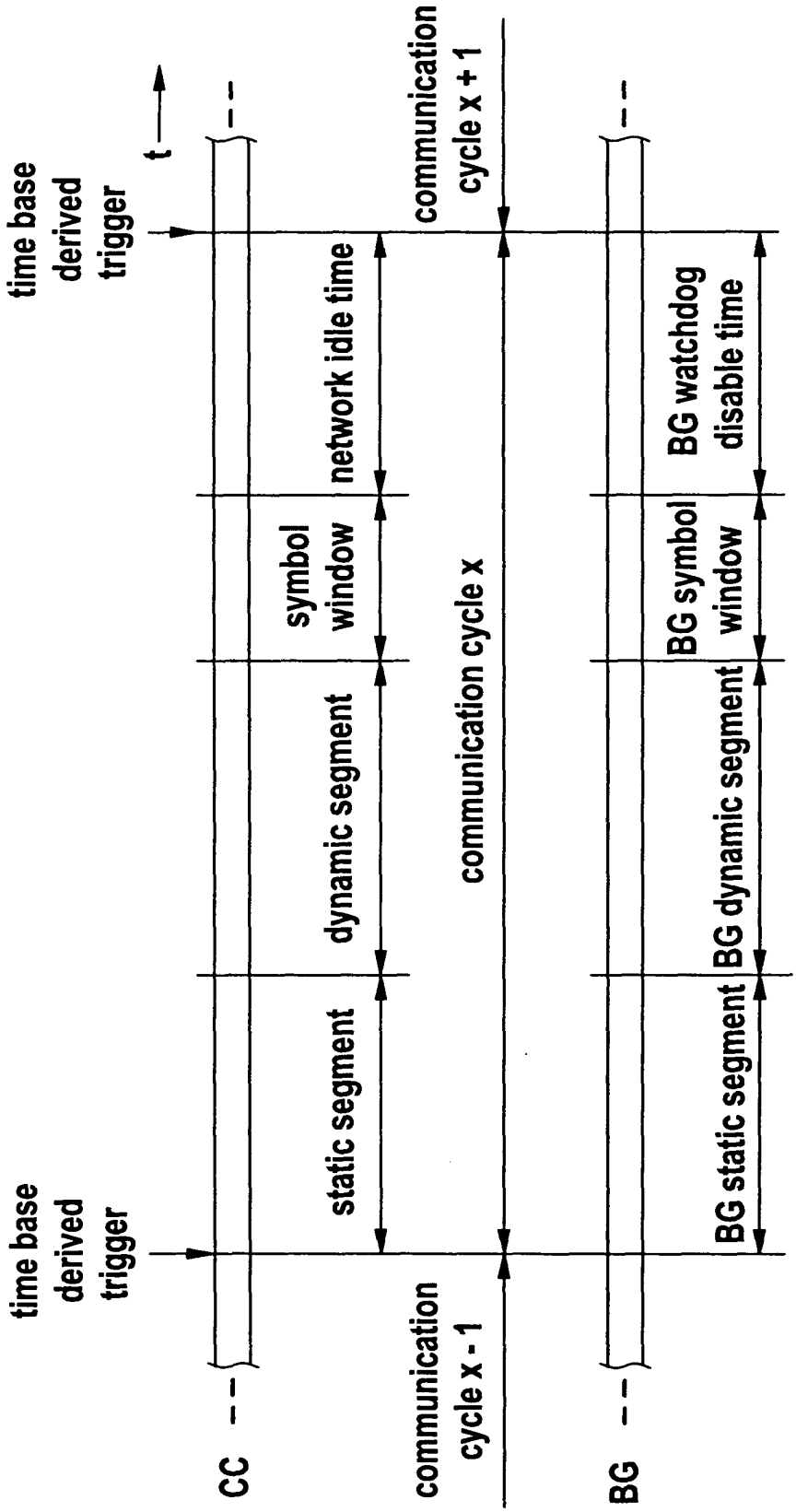

time base derived trigger
t
communication cycle x + 1
network idle time
symbol window
dynamic segment
static segment
communication cycle x
communication cycle x - 1
CC
BG
BG watchdog disable time
BG symbol window
BG dynamic segment
BG static segment

Fig. 84

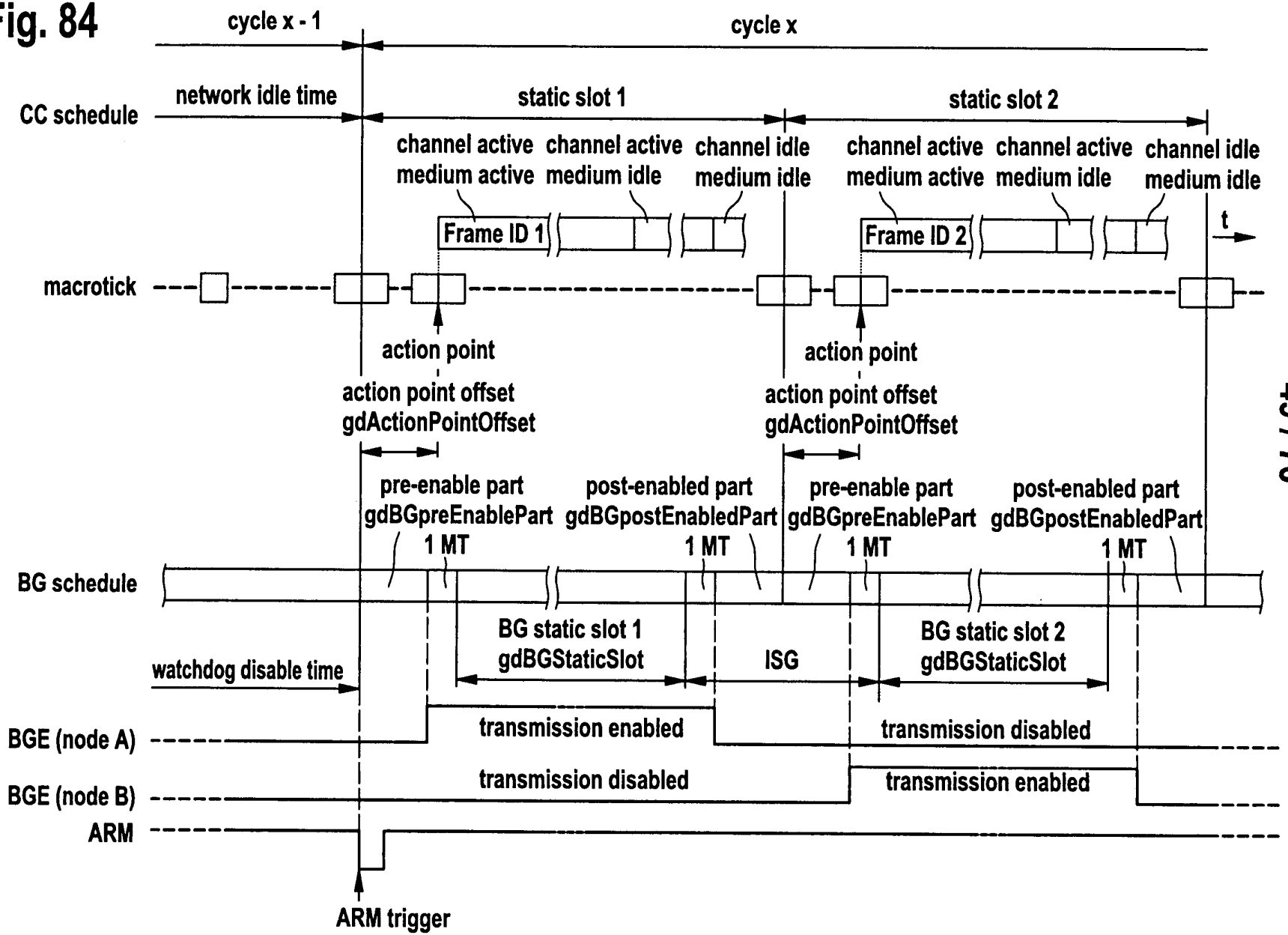

cycle x - 1
cycle x
CC schedule
network idle time
static slot 1
static slot 2
channel active
medium active
channel active
medium idle
channel idle
medium idle
channel active
medium active
channel active
medium idle
channel idle
medium idle
Frame ID 1
Frame ID 2
t
macrotick
action point
action point offset
gdActionPointOffset
action point
action point offset
gdActionPointOffset
pre-enable part
gdBGpreEnablePart
1 MT
post-enabled part
gdBGpostEnabledPart
1 MT
pre-enable part
gdBGpreEnablePart
1 MT
post-enabled part
gdBGpostEnabledPart
1 MT
BG schedule
BG static slot 1
gdBGStaticSlot
ISG
BG static slot 2
gdBGStaticSlot
watchdog disable time
BGE (node A)
transmission enabled
transmission disabled
BGE (node B)
transmission disabled
transmission enabled
ARM
ARM trigger

Fig. 85

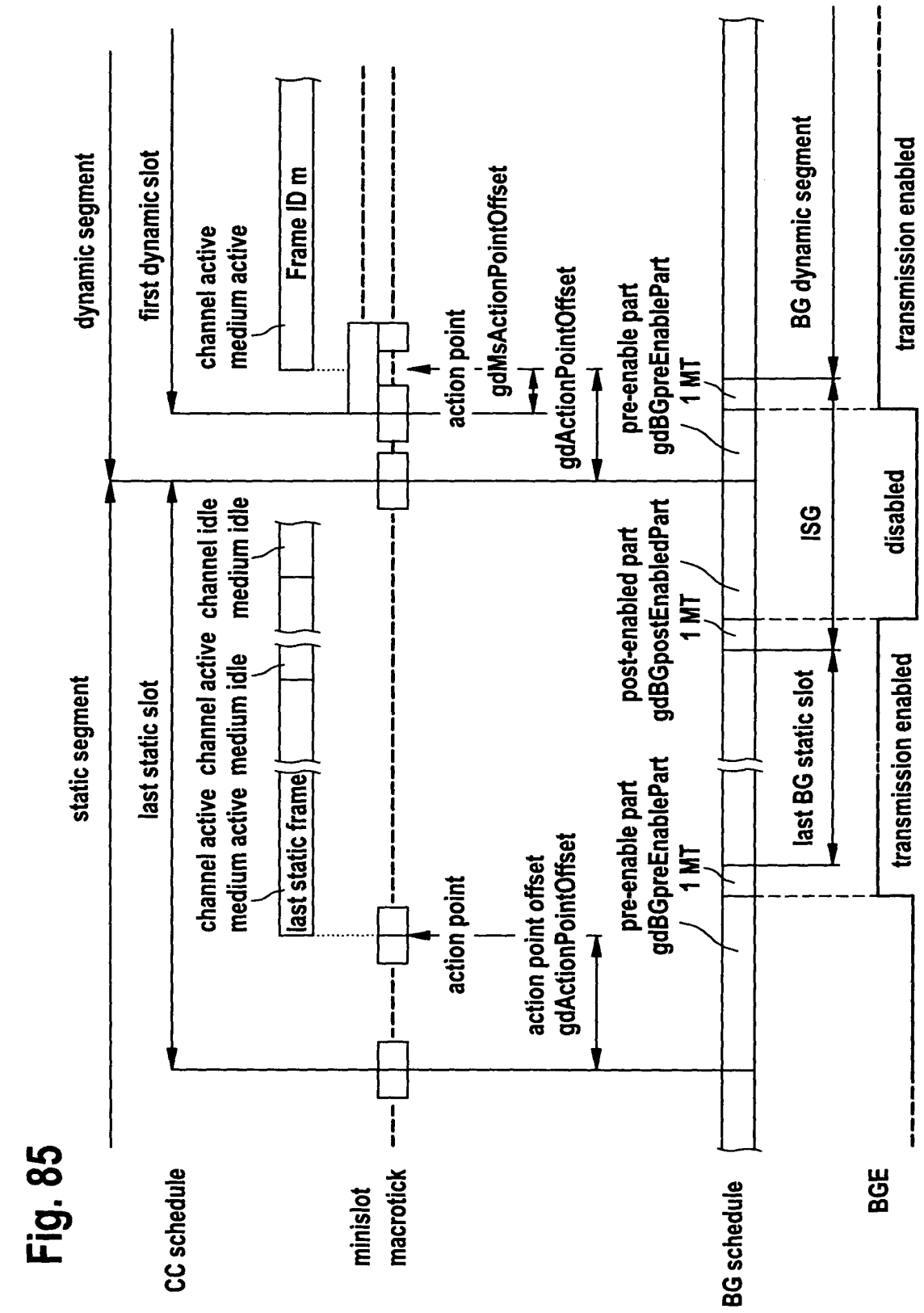

CC schedule
minislot
macrotick
BG schedule
BGE
static segment
dynamic segment
last static slot
first dynamic slot
channel active
medium active
channel idle
medium idle
last static frame
Frame ID m
action point
action point offset
gdActionPointOffset
gdMsActionPointOffset
pre-enable part
gdBGpreEnablePart
1 MT
post-enabled part
gdBGpostEnabledPart
ISG
disabled
last BG static slot
BG dynamic segment
transmission enabled

Fig. 86

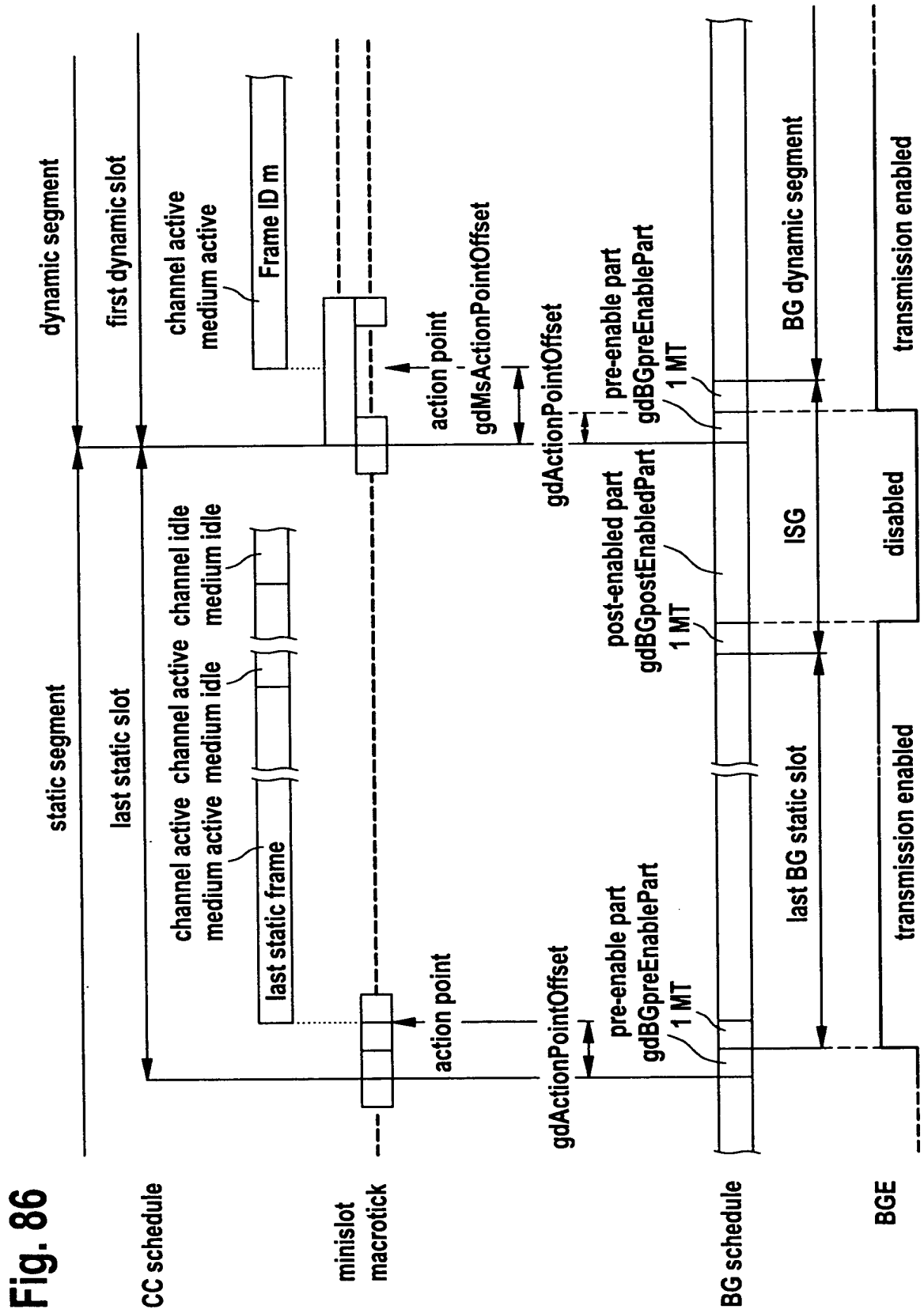

CC schedule
minislot
macrotick
BG schedule
BGE
static segment
dynamic segment
last static slot
first dynamic slot
channel active
medium active
channel idle
medium idle
last static frame
Frame ID m
action point
gdMsActionPointOffset
gdActionPointOffset
pre-enable part
gdBGpreEnablePart
1 MT
post-enabled part
gdBGpostEnabledPart
ISG
BG dynamic segment
last BG static slot
disabled
transmission enabled

Fig. 87

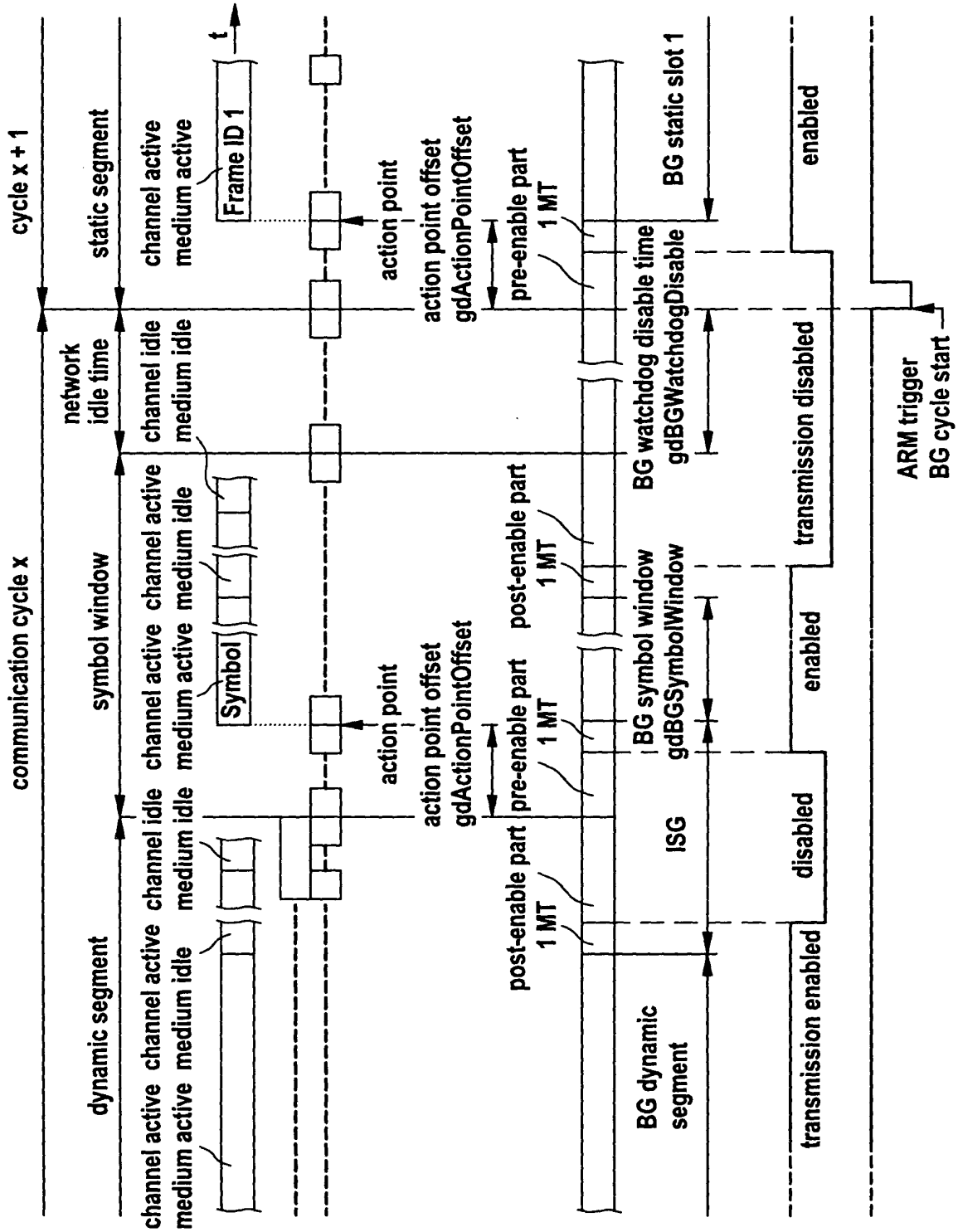

communication cycle x
cycle x + 1
dynamic segment
symbol window
network idle time
static segment
CC schedule
channel active
medium active
channel idle
medium idle
Symbol
Frame ID 1
t
minislot
macrotick
action point
action point offset
gdActionPointOffset
pre-enable part
post-enable part
1 MT
BG schedule
BG dynamic segment
ISG
BG symbol window
gdBGSymbolWindow
BG watchdog disable time
gdBGWatchdogDisable
BG static slot 1
BGE
transmission enabled
disabled
enabled
transmission disabled
ARM trigger
BG cycle start

Fig. 88

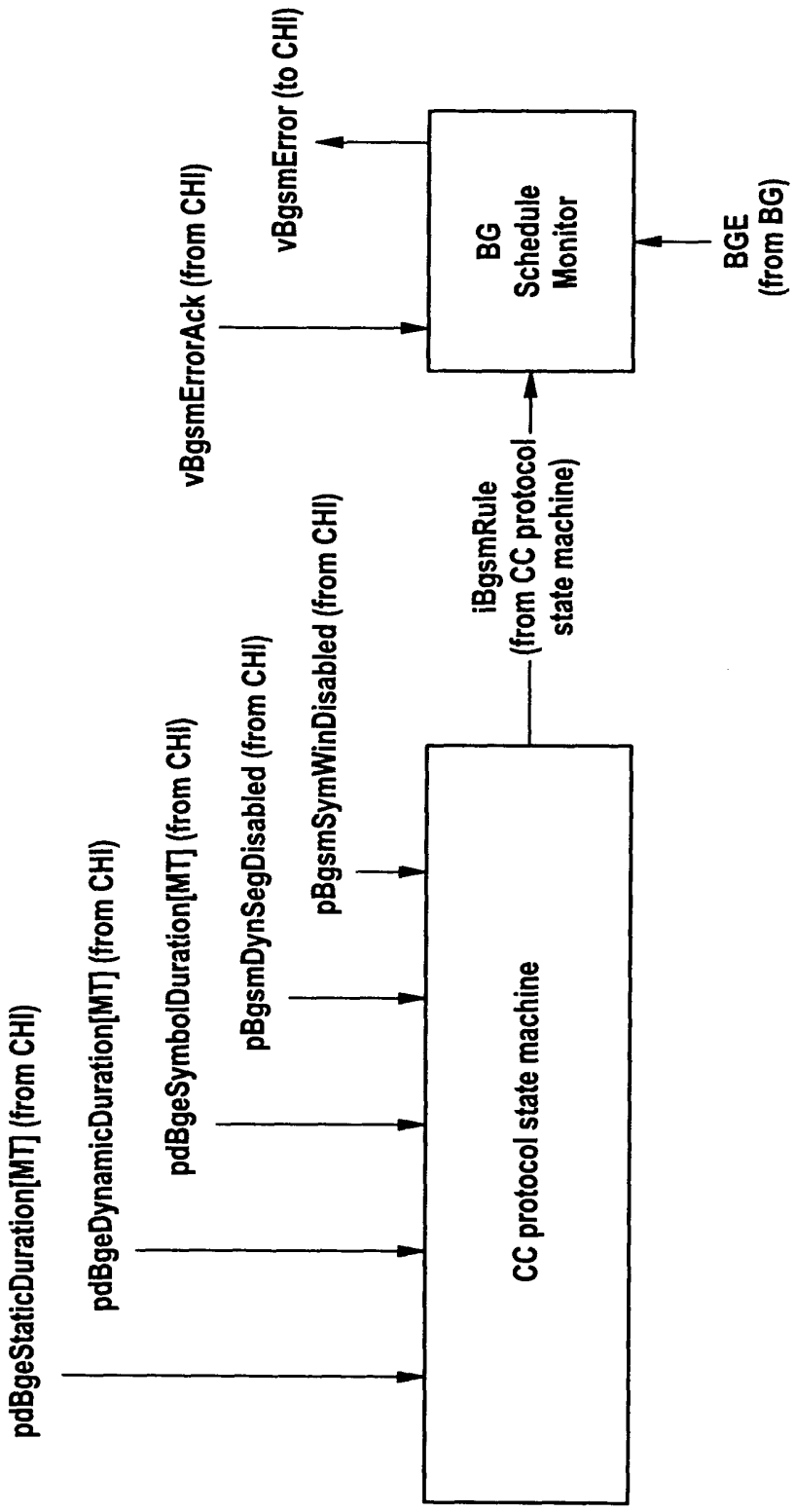

pdBgeStaticDuration[MT] (from CHI)
pdBgeDynamicDuration[MT] (from CHI)
pdBgeSymbolDuration[MT] (from CHI)
pBgsmDynSegDisabled (from CHI)
pBgsmSymWinDisabled (from CHI)
vBgsmErrorAck (from CHI)
iBgsmRule (from CC protocol state machine)
CC protocol state machine
vBgsmError (to CHI)
BG Schedule Monitor
BGE (from BG)

Fig. 89

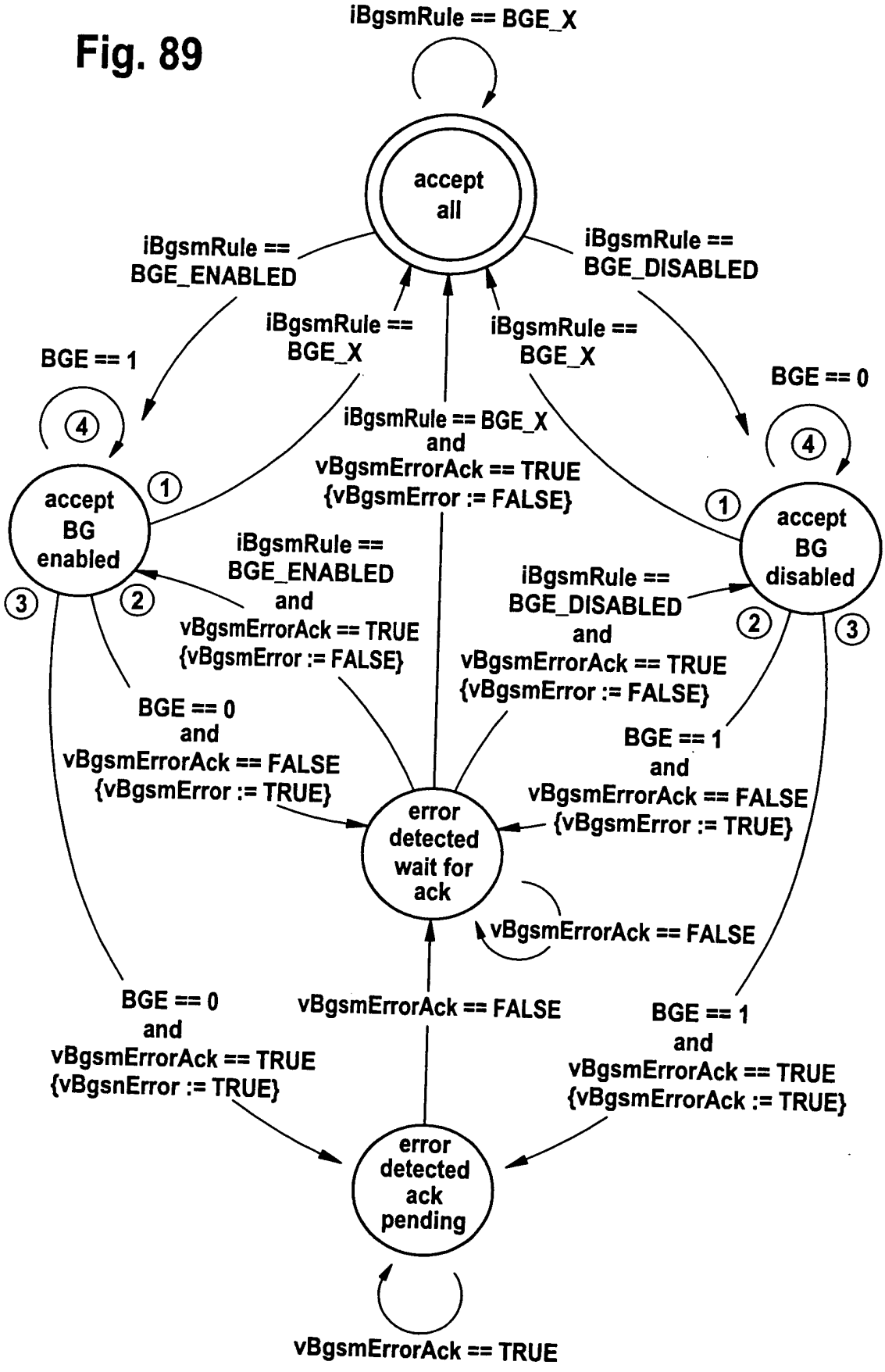

iBgsmRule == BGE_X
accept all
iBgsmRule == BGE_ENABLED
iBgsmRule == BGE_X
iBgsmRule == BGE_DISABLED
BGE == 1
BGE == 0
4
4
accept BG enabled
1
iBgsmRule == BGE_X and vBgsmErrorAck == TRUE {vBgsmError := FALSE}
1
accept BG disabled
iBgsmRule == BGE_ENABLED and vBgsmErrorAck == TRUE {vBgsmError := FALSE}
iBgsmRule == BGE_DISABLED and vBgsmErrorAck == TRUE {vBgsmError := FALSE}
3
2
2
3
BGE == 0 and vBgsmErrorAck == FALSE {vBgsmError := TRUE}
BGE == 1 and vBgsmErrorAck == FALSE {vBgsmError := TRUE}
error detected wait for ack
vBgsmErrorAck == FALSE
BGE == 0 and vBgsmErrorAck == TRUE {vBgsnError := TRUE}
vBgsmErrorAck == FALSE
BGE == 1 and vBgsmErrorAck == TRUE {vBgsmErrorAck := TRUE}
error detected ack pending
vBgsmErrorAck == TRUE

Fig. 90

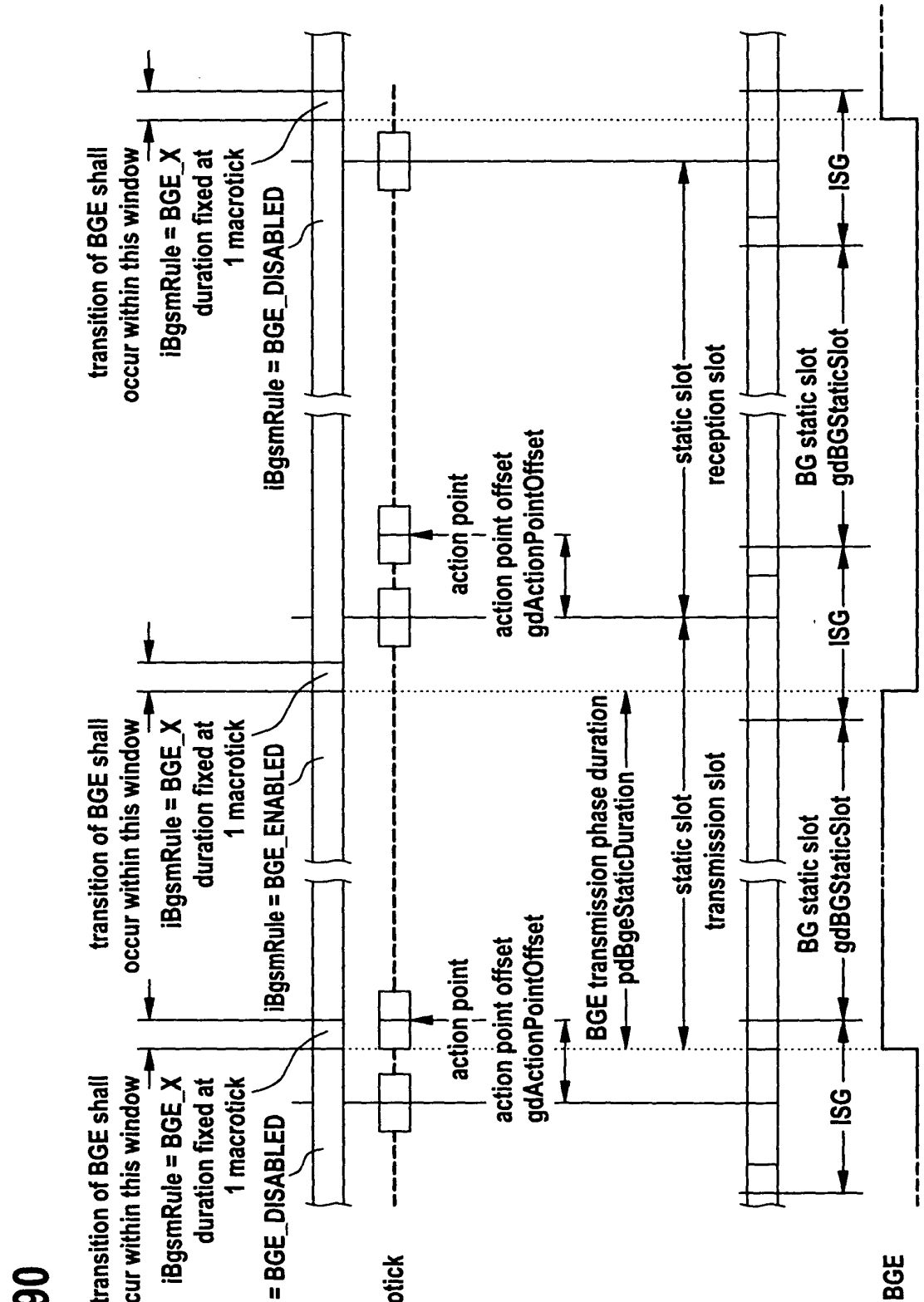

transition of BGE shall occur within this window
iBgsmRule = BGE_X duration fixed at 1 macrotick
iBgsmRule = BGE_DISABLED
iBgsmRule = BGE_ENABLED
macrotick
action point
action point offset gdActionPointOffset
BGE transmission phase duration pdBgeStaticDuration
static slot transmission slot
static slot reception slot
BG static slot gdBGStaticSlot
ISG
BGE

Fig. 91

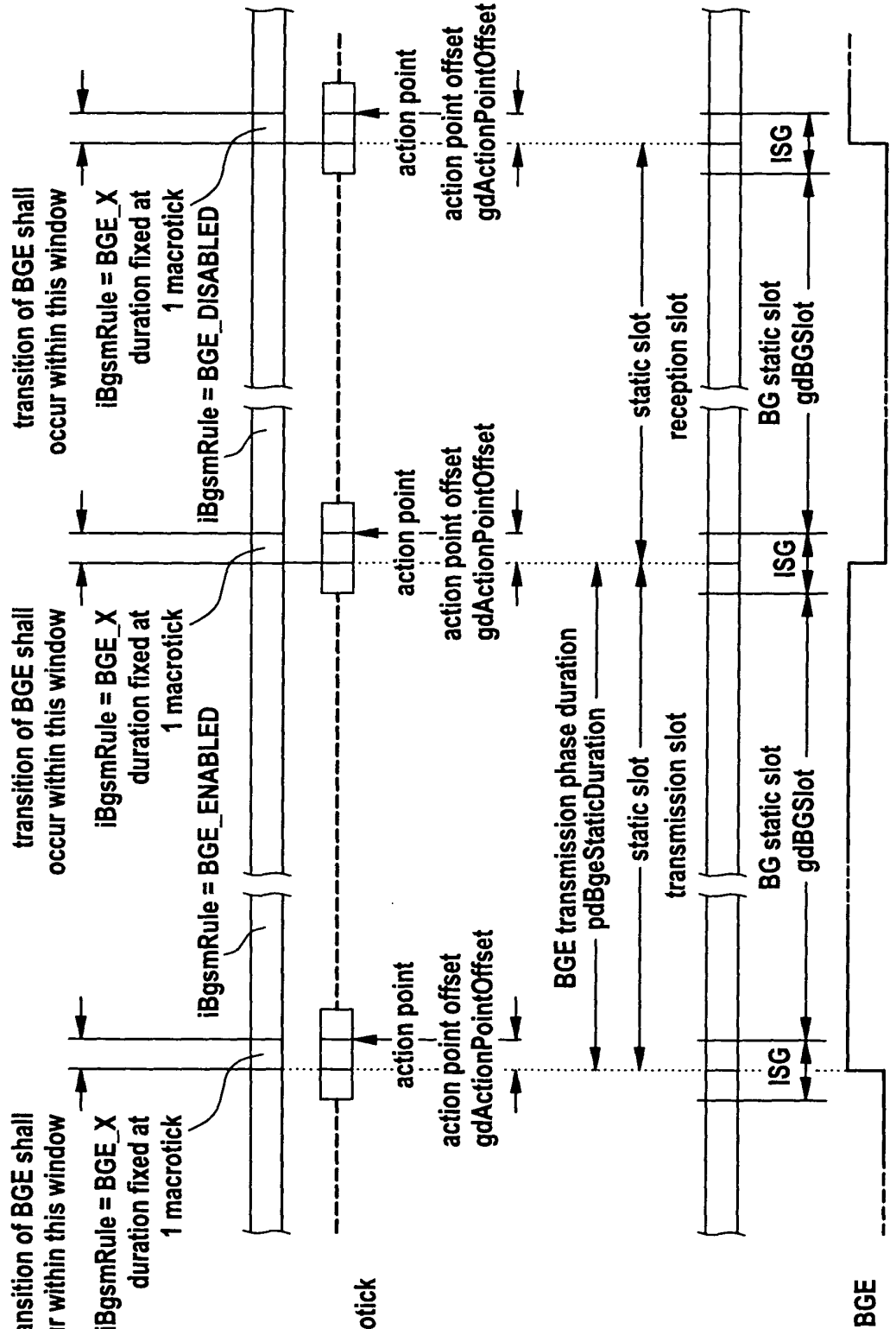

transition of BGE shall occur within this window
iBgsmRule = BGE_X duration fixed at 1 macrotick
iBgsmRule = BGE_ENABLED
iBgsmRule = BGE_DISABLED
action point
action point offset gdActionPointOffset
BGE transmission phase duration pdBgeStaticDuration
static slot
transmission slot
reception slot
ISG
BG static slot gdBGSlot
macrotick
BGE

Fig. 92

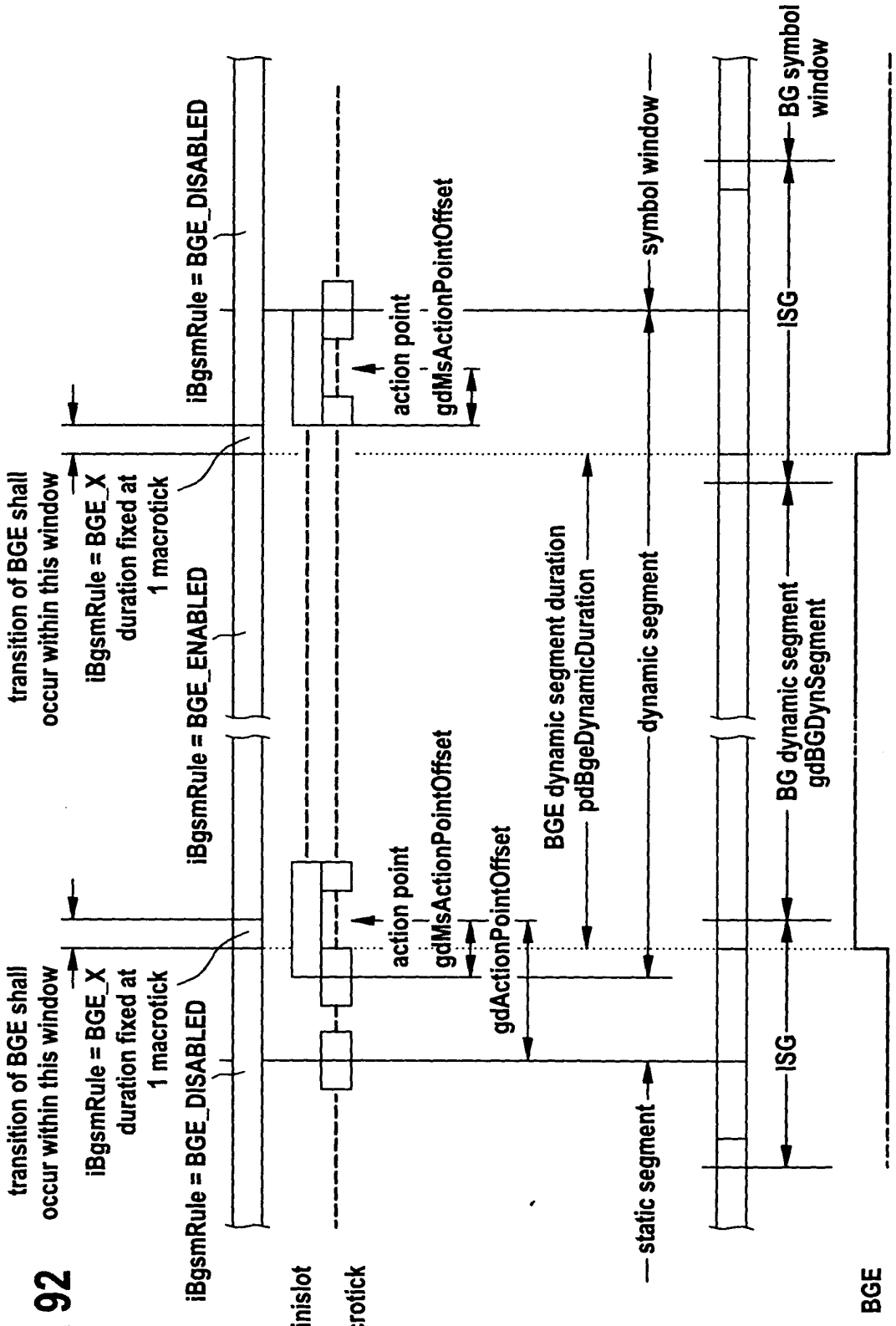

transition of BGE shall occur within this window
iBgsmRule = BGE_X duration fixed at 1 macrotick
iBgsmRule = BGE_DISABLED
iBgsmRule = BGE_ENABLED
iBgsmRule = BGE_DISABLED
action point
gdMsActionPointOffset
gdActionPointOffset
BGE dynamic segment duration
pdBgeDynamicDuration
symbol window
dynamic segment
static segment
ISG
BG dynamic segment
gdBGDynSegment
BG symbol window
minislot
macrotick
BGE

Fig. 93

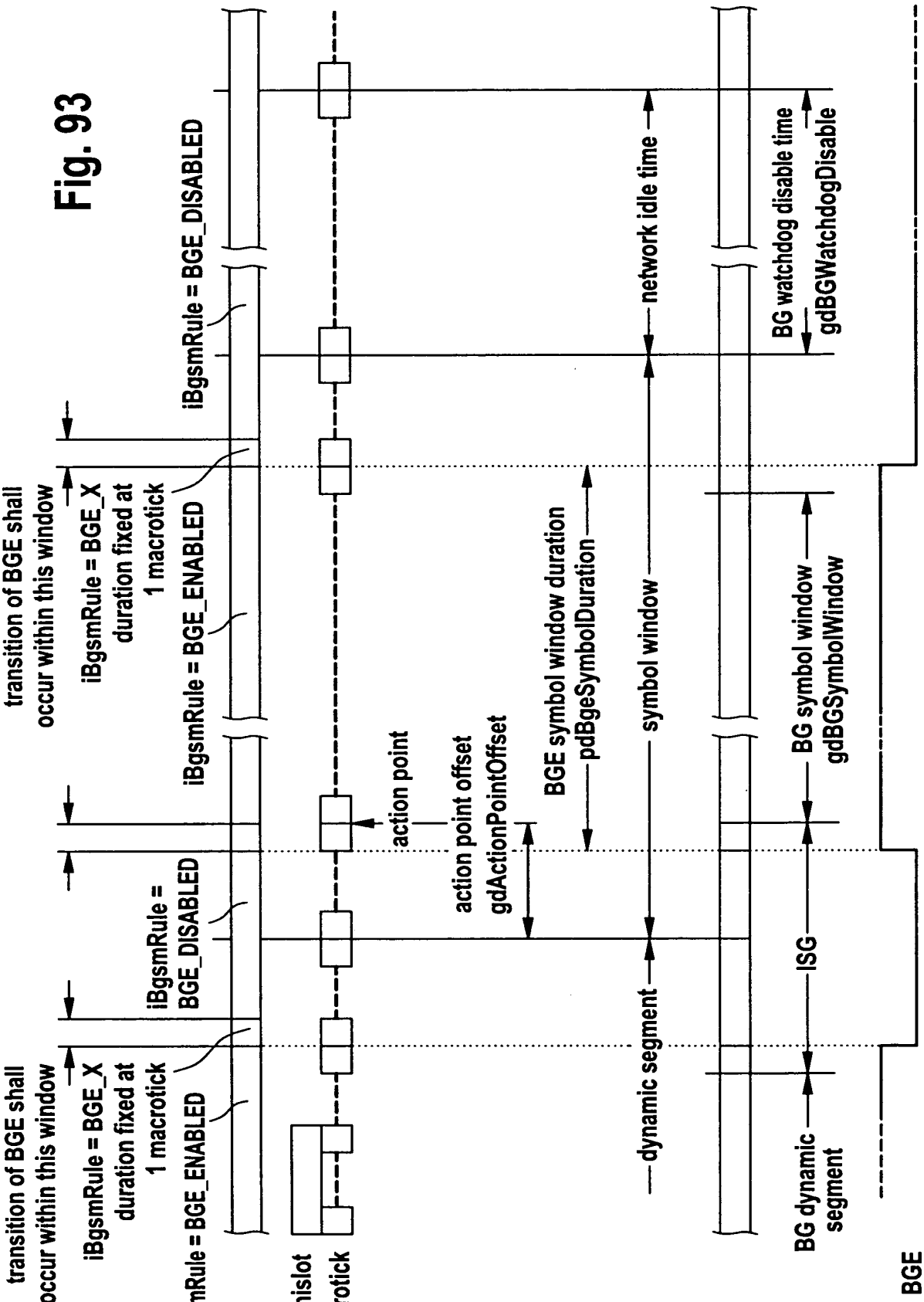

transition of BGE shall occur within this window
iBgsmRule = BGE_X duration fixed at 1 macrotick
iBgsmRule = BGE_ENABLED
iBgsmRule = BGE_DISABLED
iBgsmRule = BGE_DISABLED
iBgsmRule = BGE_ENABLED
iBgsmRule = BGE_X duration fixed at 1 macrotick
transition of BGE shall occur within this window
minislot
macrotick
action point
action point offset
gdActionPointOffset
BGE symbol window duration
pdBgeSymbolDuration
dynamic segment
symbol window
network idle time
BG dynamic segment
ISG
BG symbol window
gdBGSymbolWindow
BG watchdog disable time
gdBGWatchdogDisable
BGE

Fig. 94

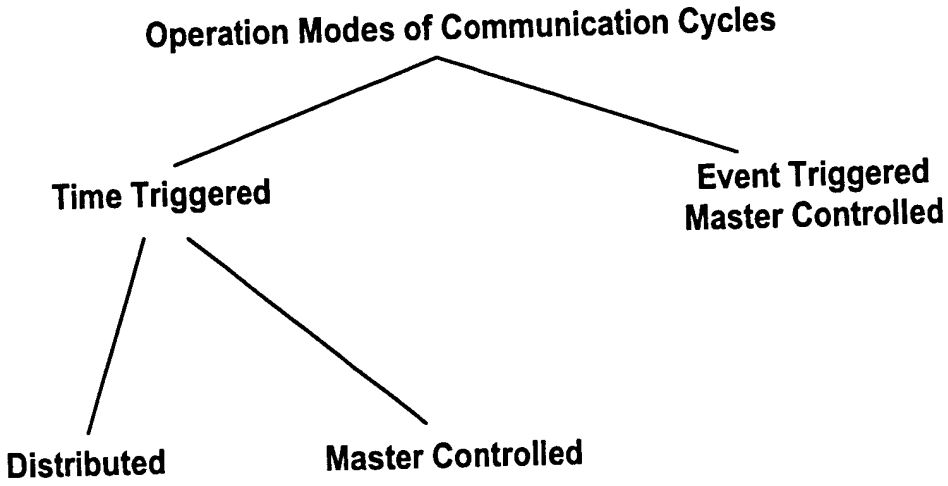

Operation Modes of Communication Cycles
Time Triggered
Event Triggered
Master Controlled
Distributed
Master Controlled

Fig. 95

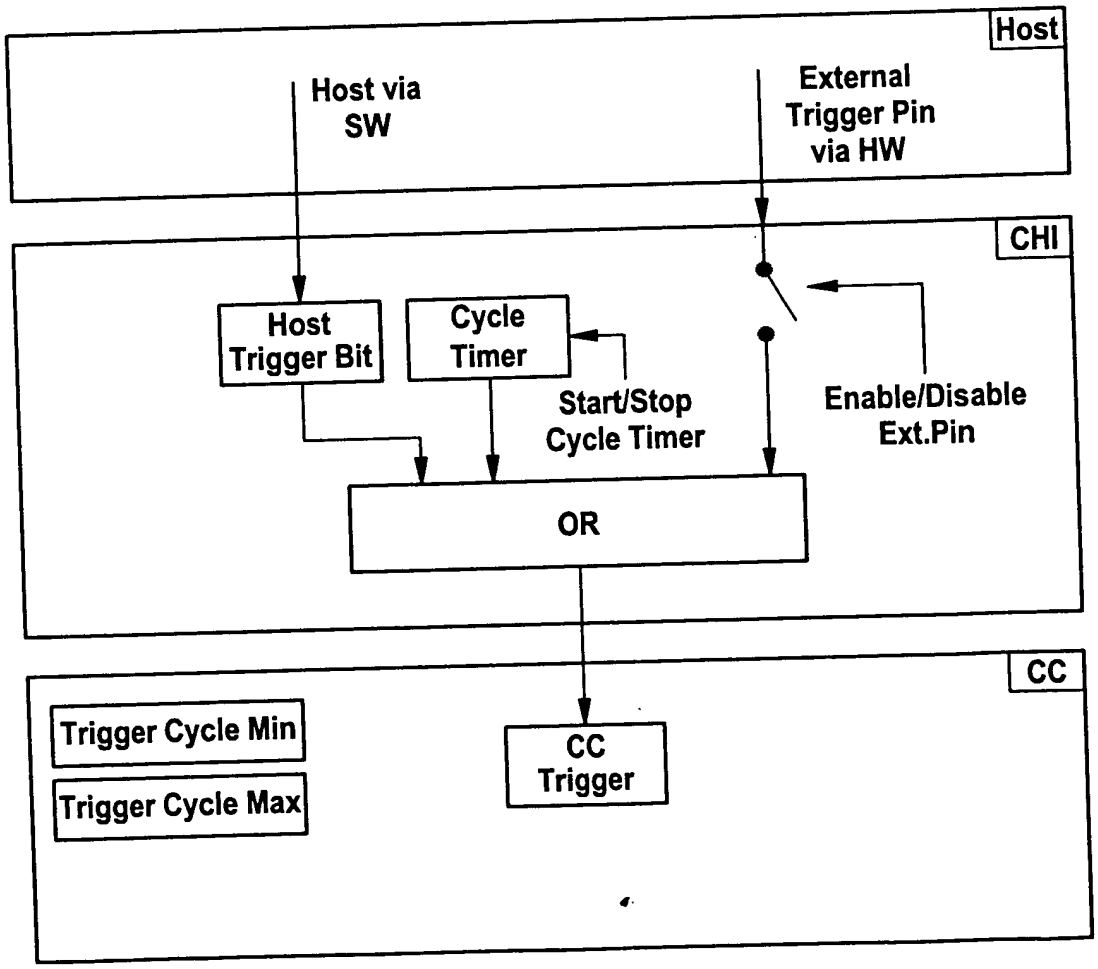

Host
Host via SW
External Trigger Pin via HW
CHI
Host Trigger Bit
Cycle Timer
Start/Stop Cycle Timer
Enable/Disable Ext.Pin
OR
CC
Trigger Cycle Min
Trigger Cycle Max
CC Trigger

Fig. 96

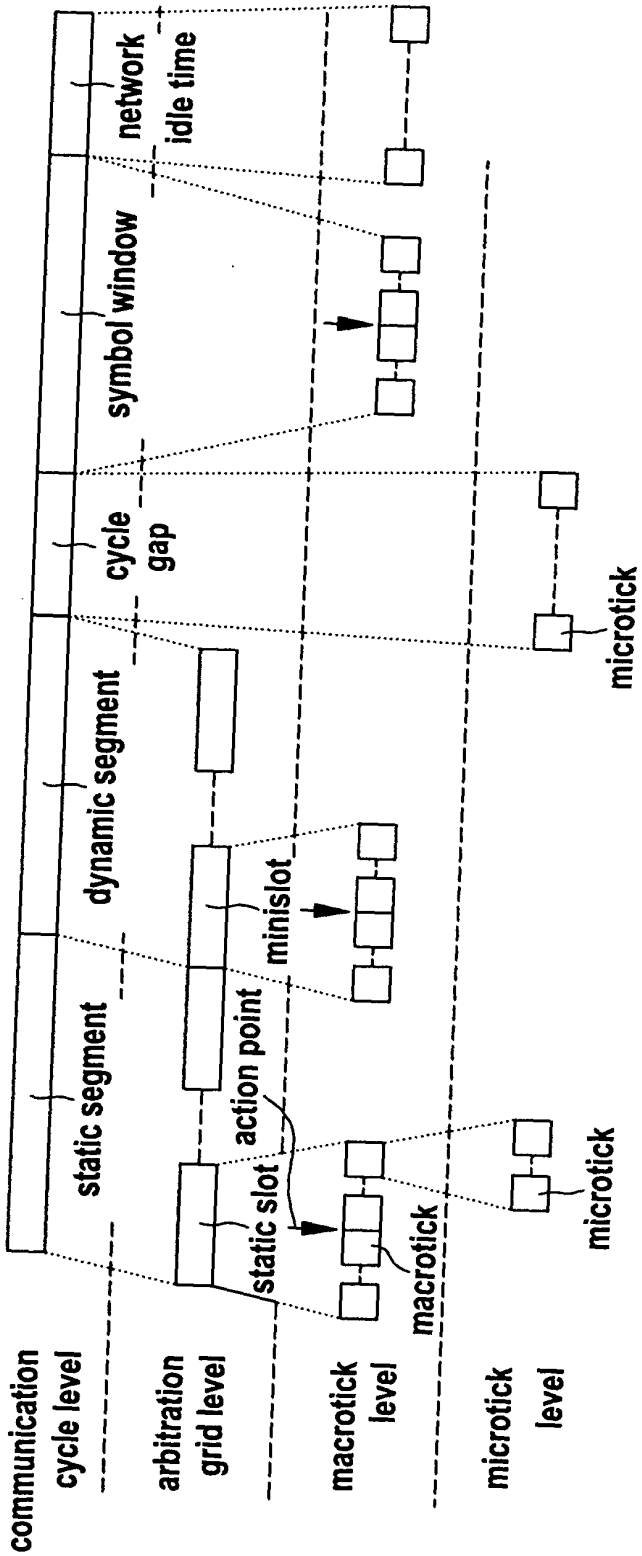

t
network idle time
symbol window
cycle gap
dynamic segment
static segment
minislot
action point
static slot
macrotick
microtick
microtick
communication cycle level
arbitration grid level
macrotick level
microtick level

Fig. 97

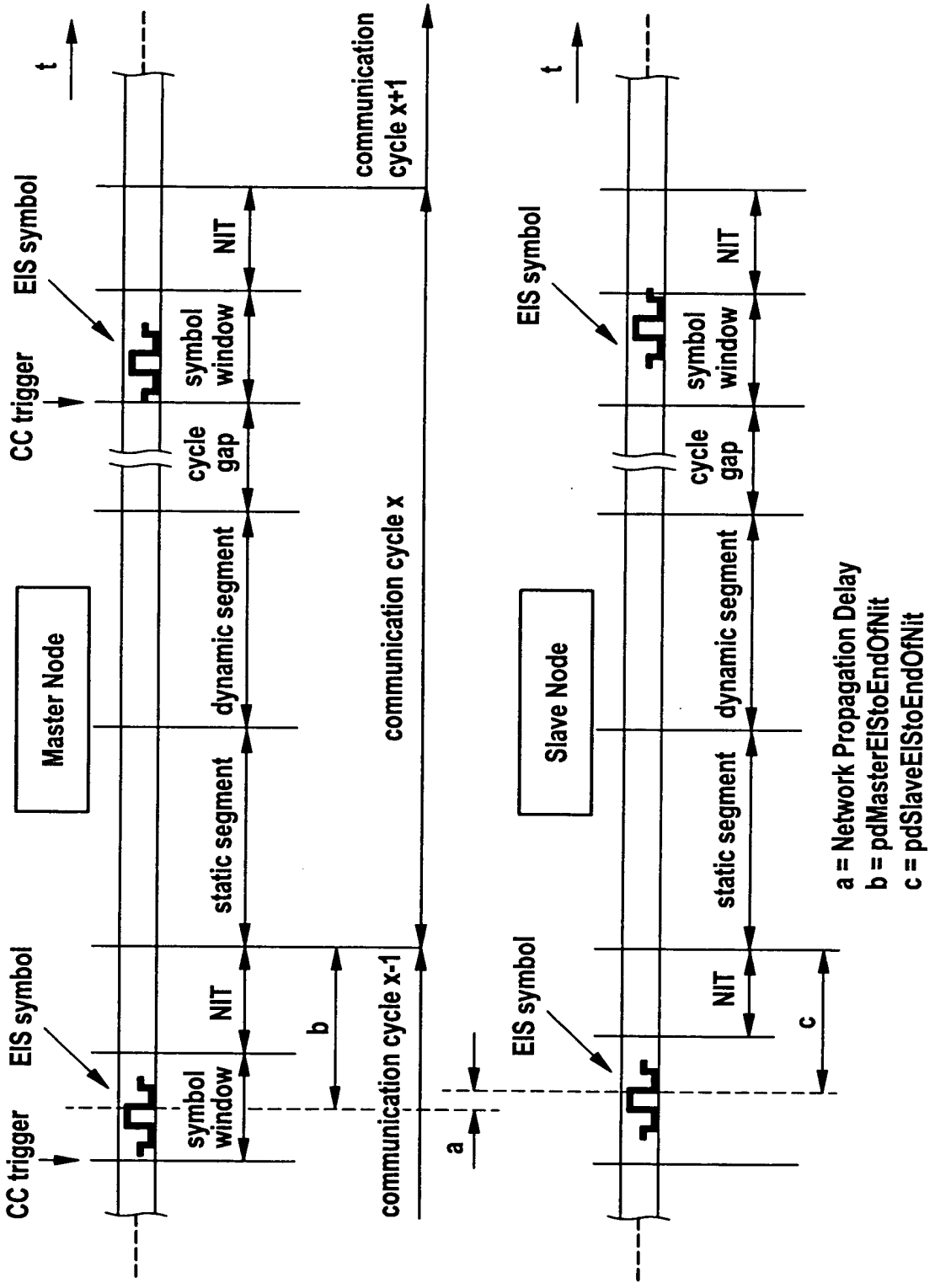

t
CC trigger
EIS symbol
symbol window
NIT
cycle gap
dynamic segment
static segment
Master Node
communication cycle x
communication cycle x+1
communication cycle x-1
b
a
Slave Node
c
a = Network Propagation Delay
b = pdMasterEIStoEndOfNit
c = pdSlaveEIStoEndOfNit

Fig. 98

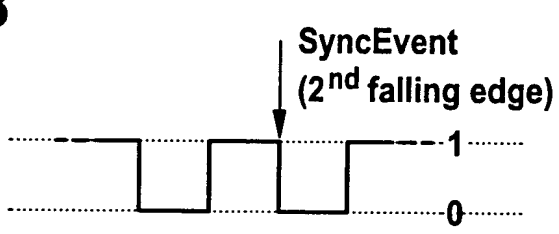

SyncEvent
(2nd falling edge)
1
0

Fig. 99

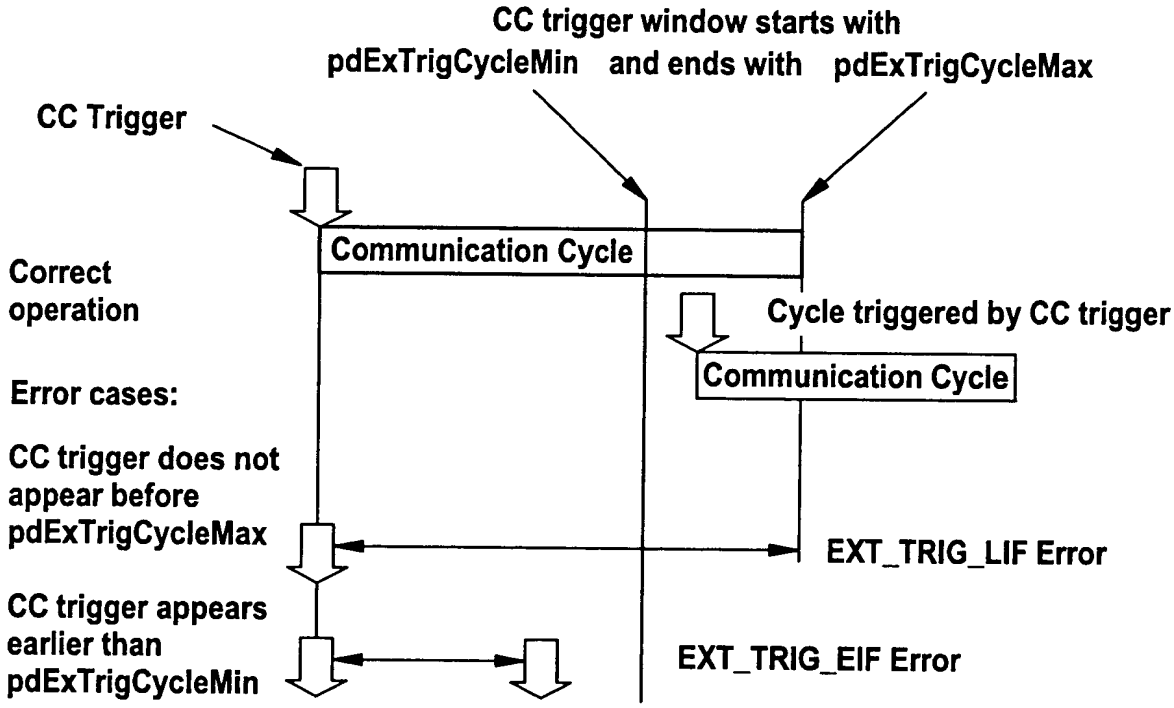

CC trigger window starts with
pdExTrigCycleMin and ends with pdExTrigCycleMax
CC Trigger
Communication Cycle
Correct
operation
Cycle triggered by CC trigger
Communication Cycle
Error cases:
CC trigger does not
appear before
pdExTrigCycleMax
EXT_TRIG_LIF Error
CC trigger appears
earlier than
pdExTrigCycleMin
EXT_TRIG_EIF Error

Fig. 100

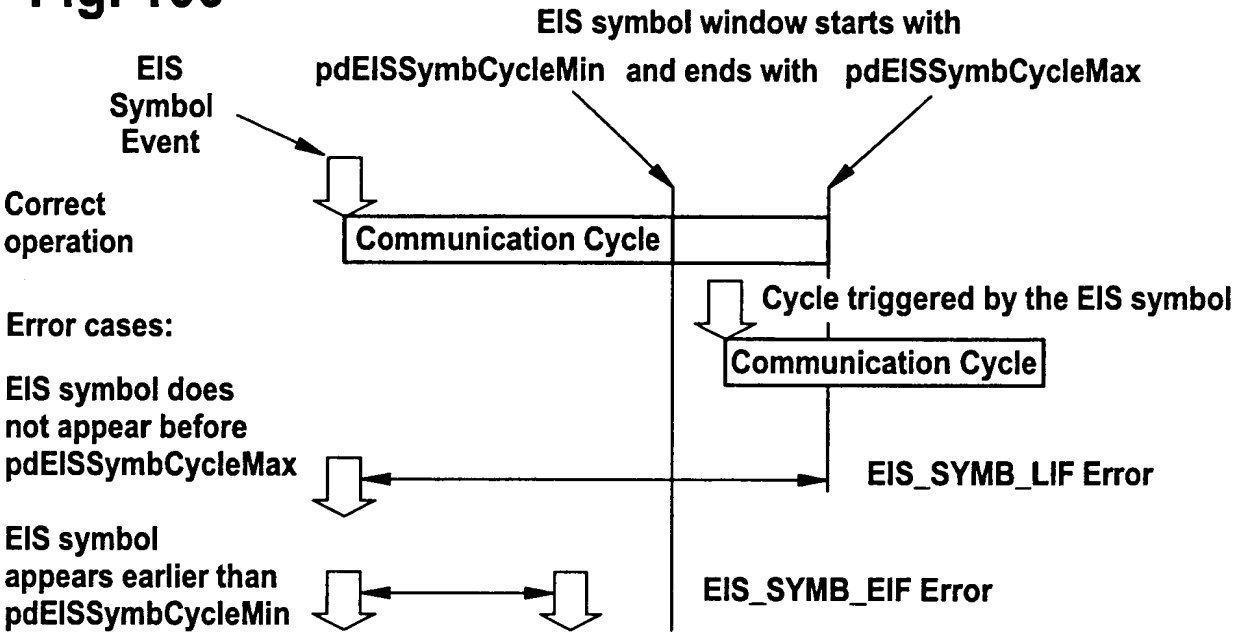


Fig. 101

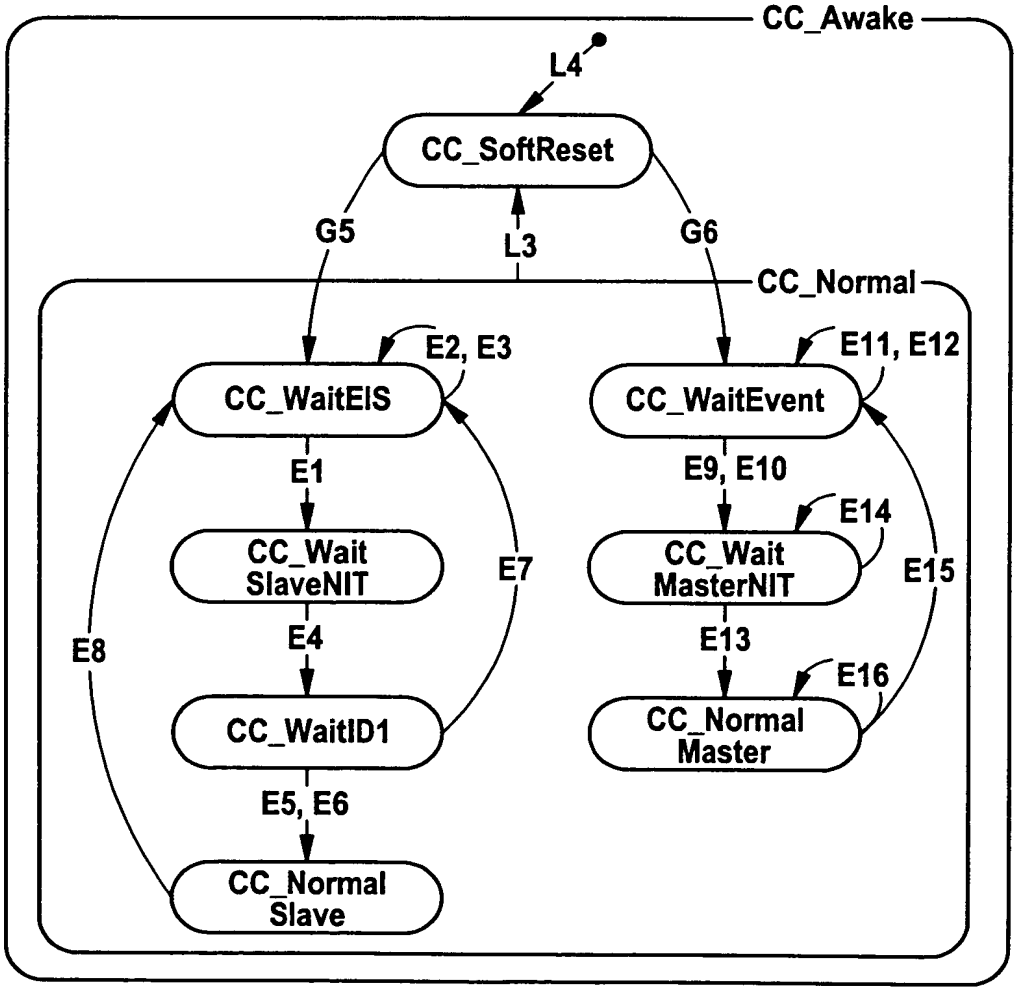


Fig. 102

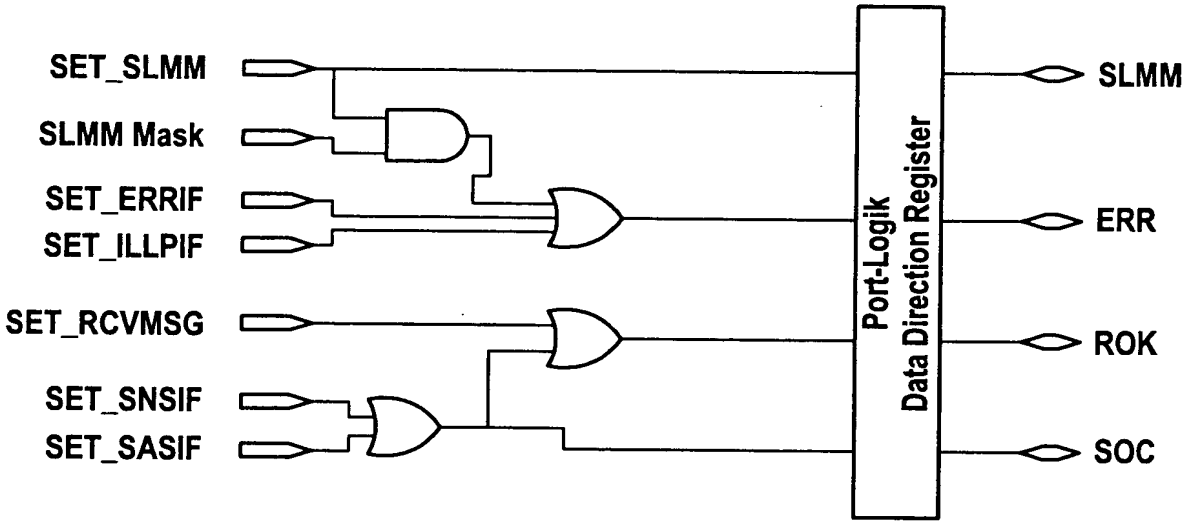


Fig. 103

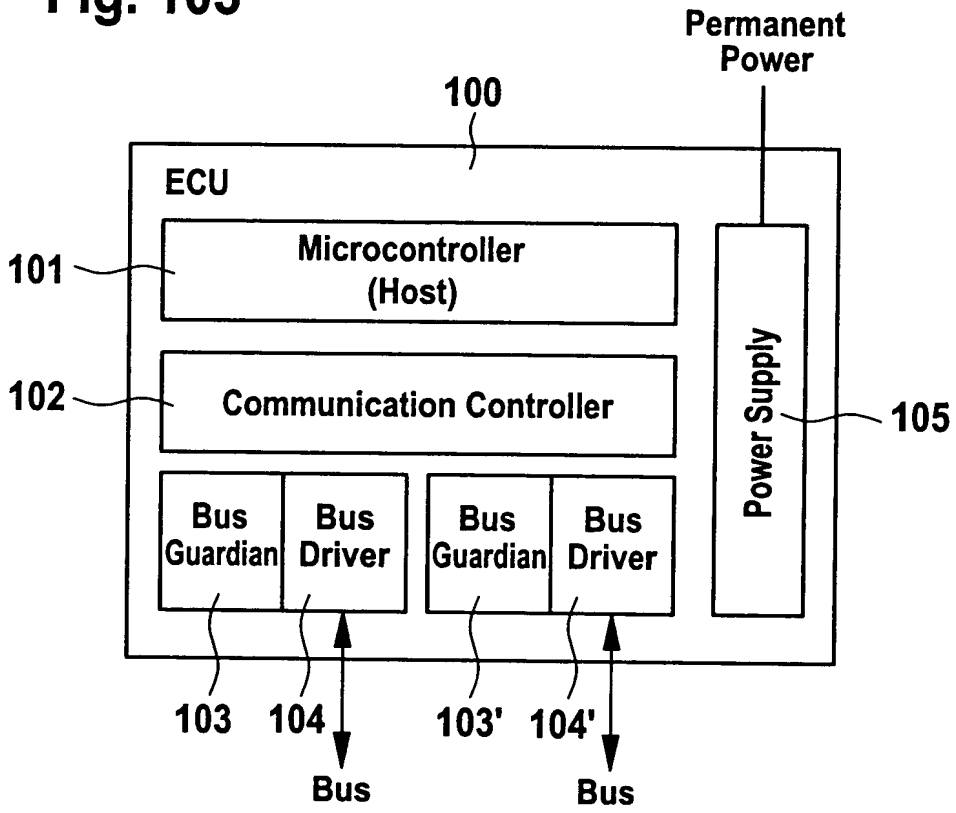


Fig. 104

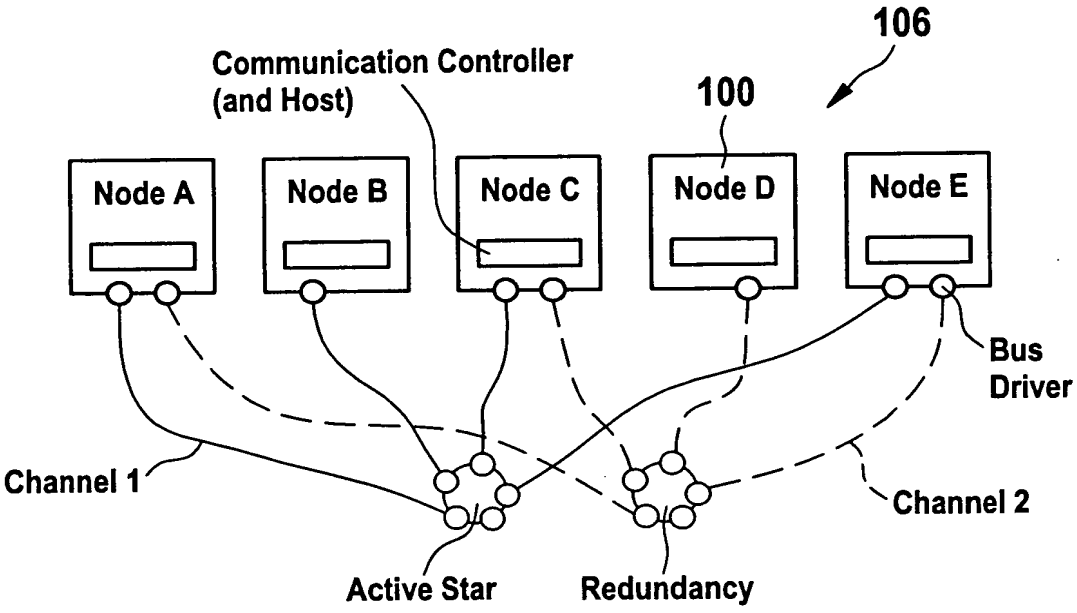


Fig. 105

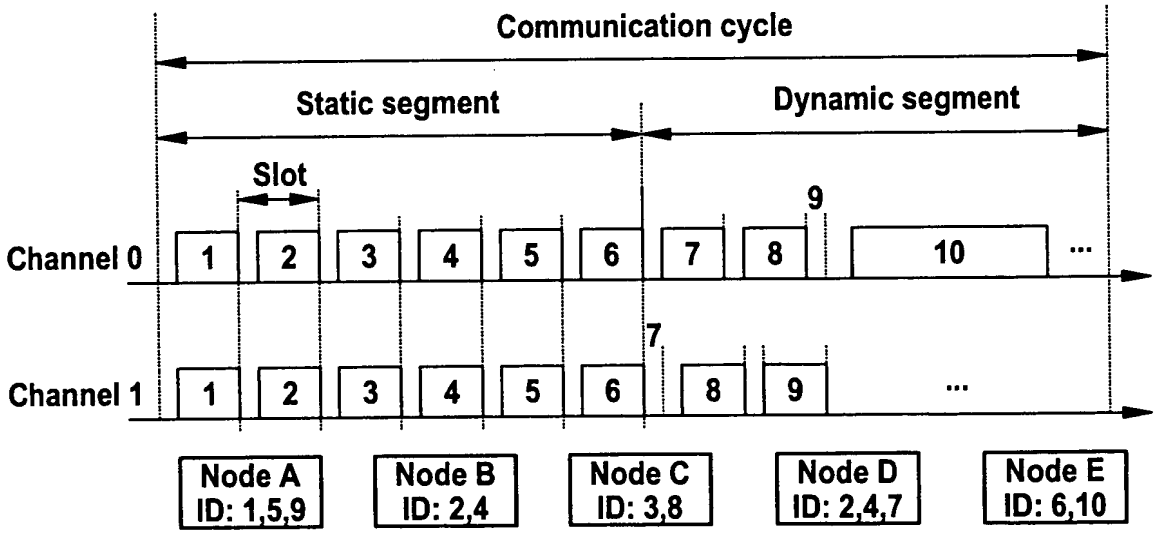


Fig. 106

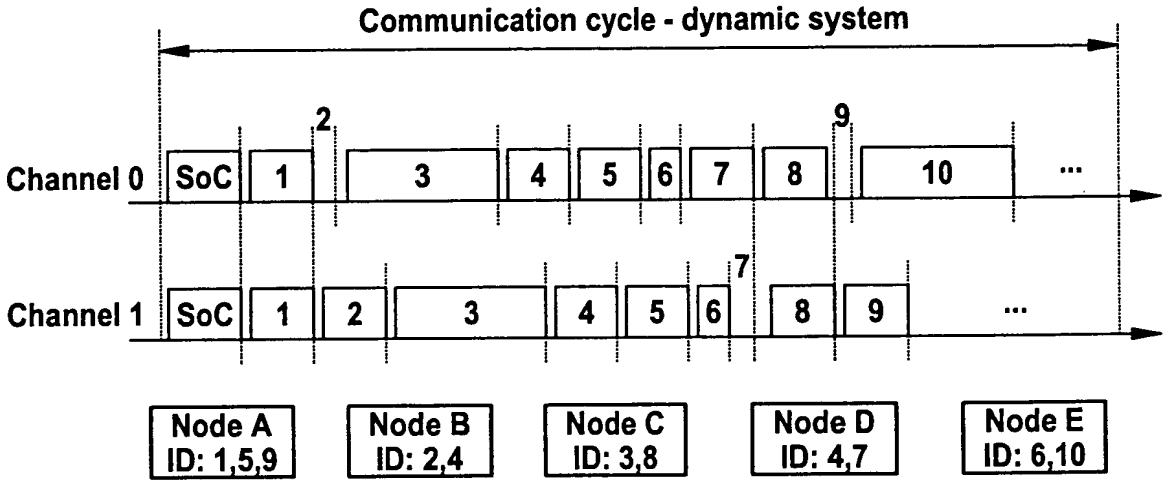


Fig. 107

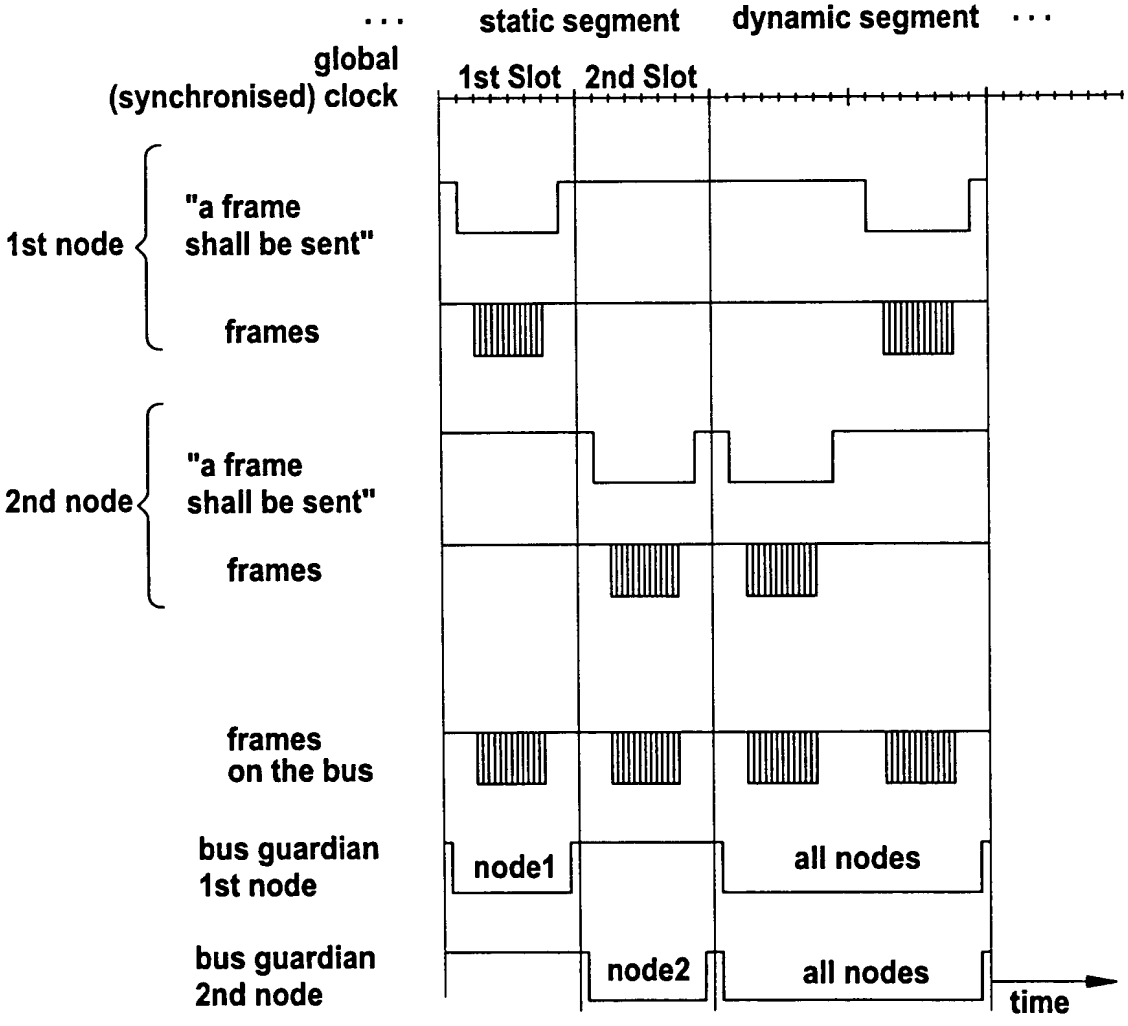


Fig. 108

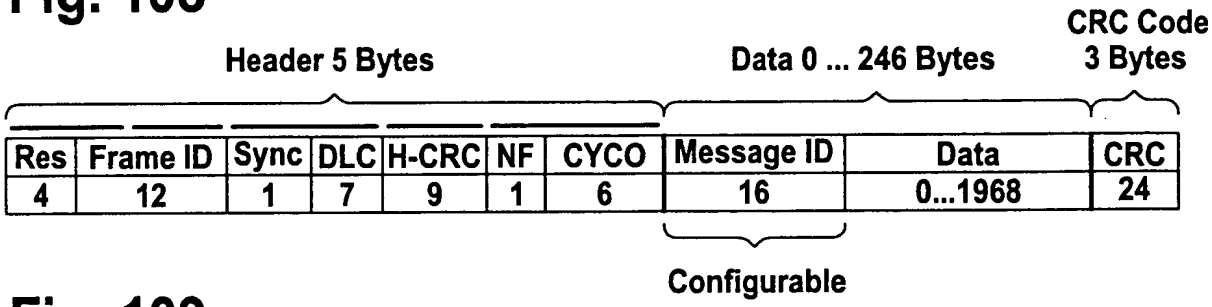


Fig. 109

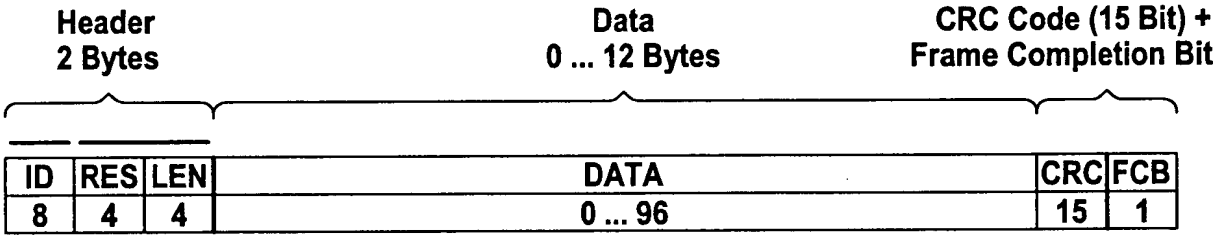


Fig. 110

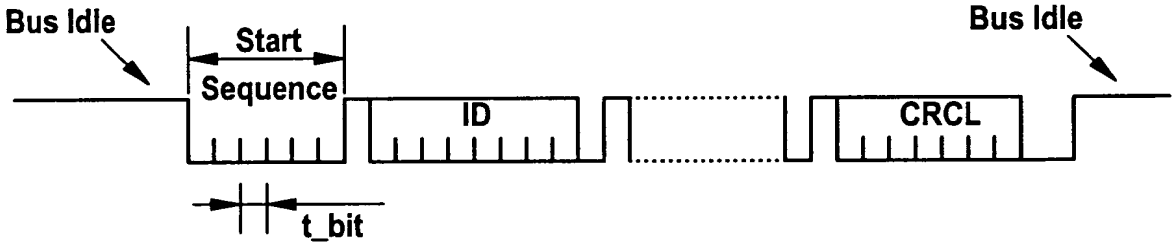


Fig. 111

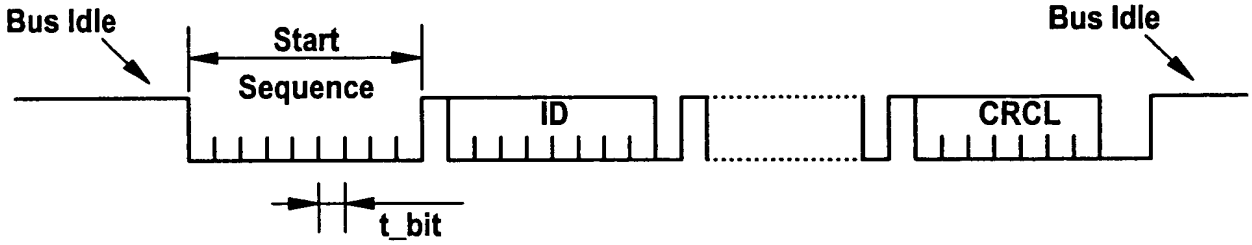


Fig. 112

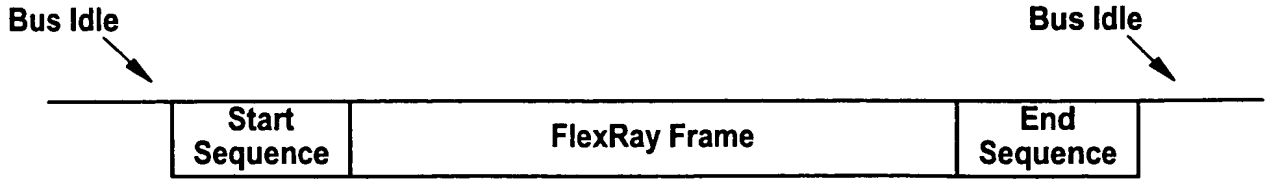


Fig. 113

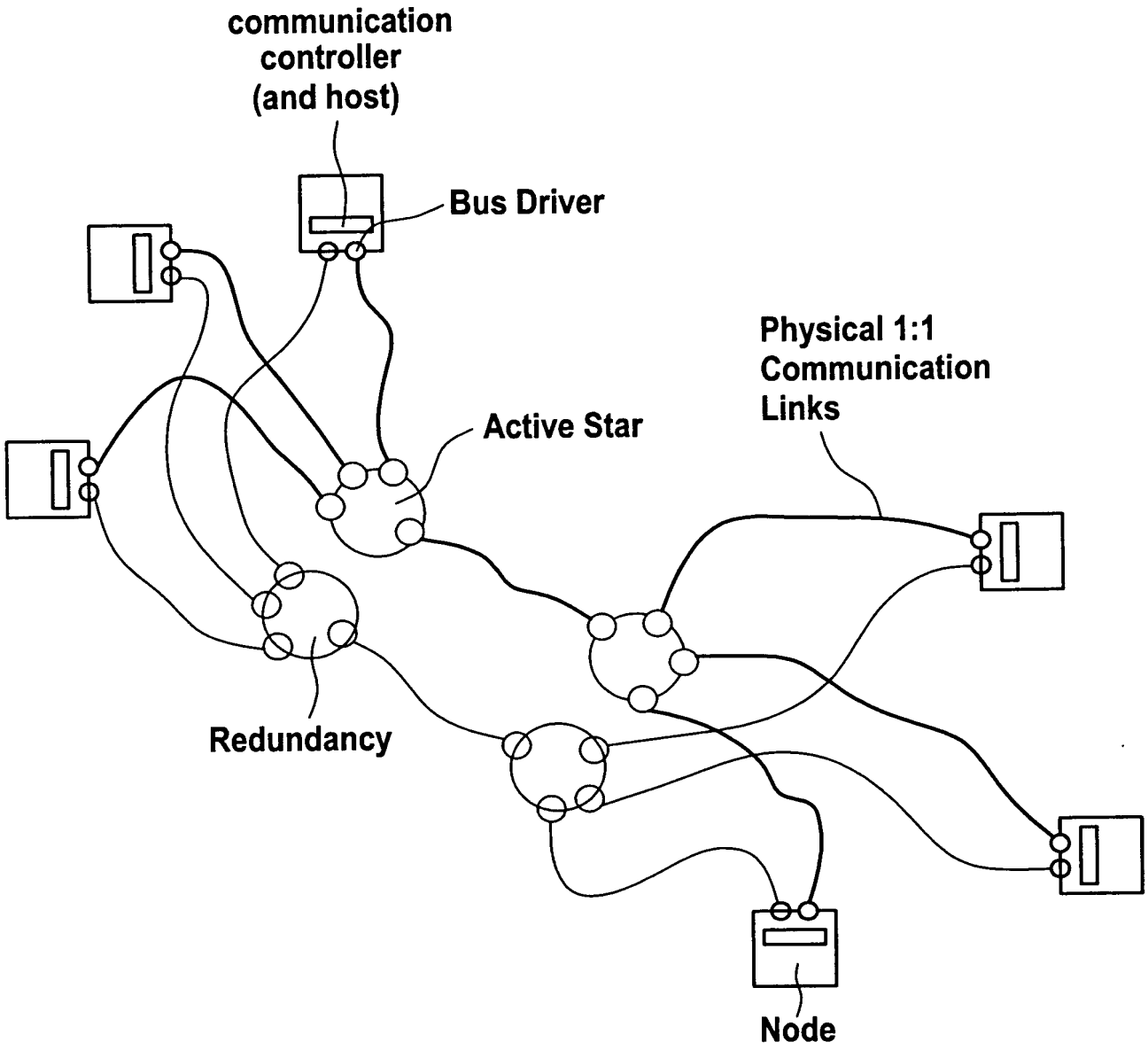

communication
controller
(and host)
Bus Driver
Physical 1:1
Communication
Links
Active Star
Redundancy
Node

Fig. 114

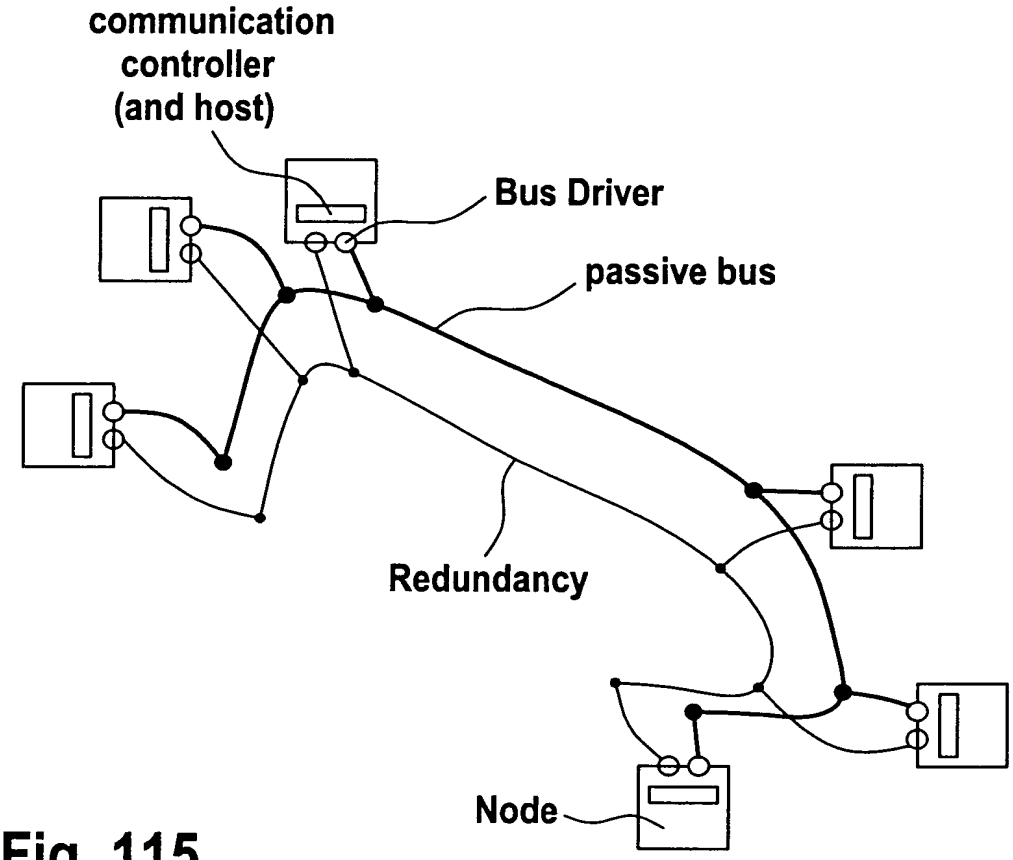


Fig. 115

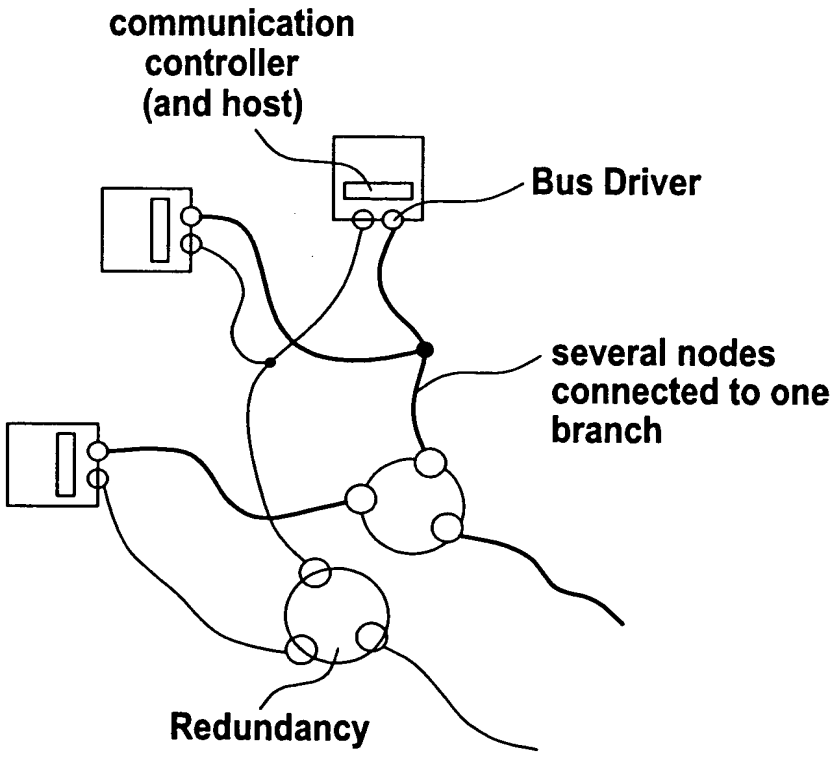


Fig. 116

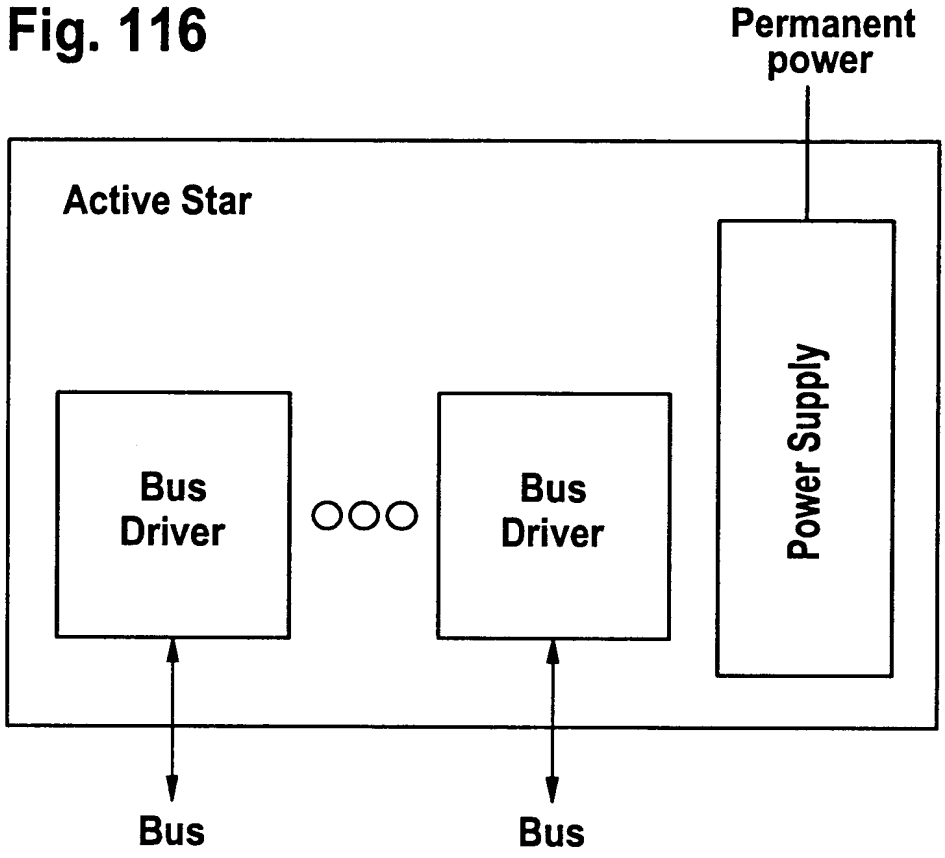
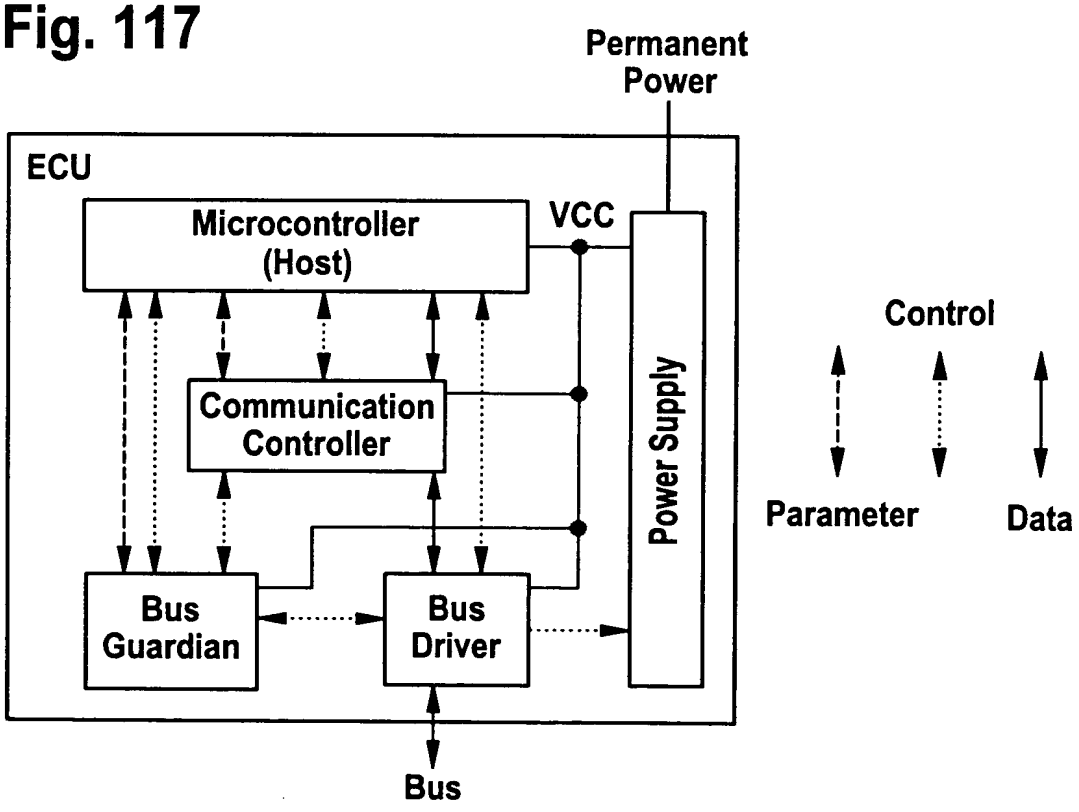



Fig. 117